

Septiembre 1984 250 ptas.
Todospectrum
AÑO 1 - NUMERO 1.
REVISTA EXCLUSIVA PARA USUARIOS
COMO USAR
EL MICRODRIVE
SINTONICE BIEN SU SPECTRUM
PROTECCION DEL SOFTWARE
LAS VARIABLES DEL SISTEMA
POSTER
GRATIS

Durante casi un año, quienes hacemos la revista ZX hemos podido comprobar la gran ilusión de todos los usuarios del Spectrum, su interés en seguir avanzando en el conocimiento de este ordenador. Para ellos nace hoy Todospectrum, una revista dedicada exclusivamente a quienes, habiendo ya adquirido suficientes conocimientos de la técnica de programación, quieres ir más allá. Esperamos que esta nueva revista merezca de sus lectores la misma calurosa acogida que ha recibido ZX. Y, desde luego, estas páginas están abiertas a la colaboración de todos.





El dibujo de la portada ha sido realizado con el programa Artist por Fernando García, en base a un diseño de Alberto Fregenal.

Todospectrum es una publicación de Publinformática, S.A., C/Bravo Murillo, 377 - Madrid 28020. Tel. (91) 733 74 13 / 47 / 63 / 97

REDACCION
Director:
Simeón Cruz.
Colaboradores:
Juan Arencibia.
Fernando García.
Antonio Lenguas.
Manuel Arias.
Gumersindo García.
Diseño:
Ricardo Segura.

EDITORIAL
Presidente:
Fernando Bolín.
Director Editorial:
Norberto Gallego.
Coordinador Editorial:
J. A. Sanz.

ADMINISTRACION
Gerente de Circulación y Ventas:
Luis Carrero.
Producción:
Miguel Onieva.
Publicidad Madrid:
Bravo Murillo, 377.
Madrid 28020.
Publicidad Barcelona:
Pelayo, 12.
Tel. (93) 301 47 00, ext. 27.
Distribuye:
S.G.E.L.
Avda. Valdelaparra, s/n.
Alcobendas, Madrid.
Imprime:
Héroes. C/Torrelara, 8.
Madrid 28036.
Fotomecánica:
Karmat. Pantoja, 10. Madrid.
Depósito Legal: M 29401 - 1984

CÓMO USAR EL MICRODRIVE

Con el atractivo y ambiguo nombrede *microdrive*, Sinclair lanzó definitivamente al mercado una unidad de almacenamiento de datos para el Spectrum a finales del año pasado. Las previsiones de ventas quedaban claras en el anuncio de la famosa empresa en Inglaterra: "Se entregarán de acuerdo a la fecha de compra de los Spectrums y no más de dos unidades por persona". Aquí, como siempre, tuvimos que esperar más. A finales de mayo de este año salía definitivamente a la venta a un precio poco asequible para los jóvenes usuarios del Spectrum.

Para poder utilizarlo, como ocurre con los *joystick* y cualquier periférico en general, necesita de un *interface*. Sinclair creó especialmente el *interface* 1 para este propósito, pero no es ésta su única característica. Conviene, pues, analizar por separado ambos productos que podrá igualmente adquirir por separado (lógicamente sólo necesitará un *interface* pero puede disponer de varios *microdrives)*.

EL MICRODRIVE

Cada *microdrive* puede albergar una pequeña cinta de las denominadas "sin fin". La capacidad de esta cinta es de 100 K sin formatear, que se quedan aproximadamente en 80 una vez formateada (operación que requiere treinta segundos). El tiempo medio de acceso es de 3,5 segundos. El número máximo de unidades que se pueden conectar es de ocho, por lo que se puede disponer de algo más de 640 K accesibles de forma inmediata, pero no simultánea. Al igual que con el *cassette*, puede intercambiar cintas para disponer de mayores capacidades.

Las ventajas respecto al *cassette* son claras: mayor capacidad y rapidez en el acceso a la información. Pero dista mucho de tener las posibilidades de un disco. Permite trabajar con archivos de datos, pero únicamente del tipo secuencial: no existen registros. En suma, no olvide que es una cinta a pesar de su nombre.

Para grabar programas es ideal por su rápida grabación y localización. También existe la posibilidad de pedir el catálogo de los programas o archivos contenidos en una cinta, indicando el nombre de la cinta, información contenida en ella y espacio disponible en K. Para trabajar con archivos de datos es igualmente útil, siempre que la grabación se realice en bloque, es decir, de una sola vez; de lo contrario perderá mucho tiempo. Ello es debido a que no se puede grabar sobre un archivo ya creado, por lo que toda actualización de datos implica la creación de un nuevo archivo para pasar los datos antiguos, añadir los nuevos y eliminar el anterior archivo. Como se puede ver, sustancialmente distinto al funcionamiento de un disco. Cada cinta puede almacenar 50 archivos como máximo.

El motor de arrastre de la cinta sólo entra en funcionamiento cuando se hace uso de ella, indicándolo con el encendido de un diodo LED. De esta forma se evitan problemas de calentamiento. La cinta sólo se podrá introducir o extraer cuando el diodo esté apagado, recomendándose colocarla una vez conectado el Spectrum a la alimentación y retirándola antes de su desconexión de la red.

No obstante, es un gran avance comparado con el *cassette* y a buen precio comparado con los discos, clave de todos los productos Sinclair (un *microdrive* cuesta 19.500 pesetas y el *interface* otras 19.500 pesetas). El precio de una unidad de discos no baja de 80.000 pesetas. Sin embargo, las cintas resultan demasiado caras: 1.650 frente a las 500 pesetas. de los convencionales discos de cinco pulgadas, amén de su enorme desgaste que le obligará a constantes copias de sus datos para evitar su pérdida. Aunque se habla de 5000 accesos mínimos, la práctica ha demostrado una duración menor.

FICHA

() MICRODRIVE
— Número máximo de unidades a conectar simultáneamente, 8.
— Tiempo medio de acceso a la información, 3,5 seg.
— Precio, 19.500 ptas.

() CINTA
— Longitud, 15 metros.
— Capacidad, 85 K.
— Número máximo de archivos, 50.
— Precio, 1.650 ptas.

() INTERFACE 1
— Salida expansión.
— Salida para red de área local (hasta 64 Spectrums).
— Salida RS232.
— Controlador *microdrive* (hasta 8).
— Precio: 19.500 ptas.

Sinclair ha prometido lanzar *software* utilizando esta unidad de almacenamiento, especialmente aplicaciones profesionales. Si pensamos que el nuevo ordenador de Sinclair (QL o *quantum leap)* utiliza este mismo sistema es fácil esperar desarrollos futuros.

INTERFACE 1

El diseño del *interface* 1 es particularmente atractivo al ajustar perfectamente en la ranura de expansión y elevar el ángulo del teclado para trabajar más cómodamente. Dos tornillos en la parte inferior permiten una segura fijación. Por la parte posterior dispone de una ranura de expansión, red local y salida RS232. En el lado izquierdo encontramos un nuevo *port* para el control de los *microdrives.*

La ranura de expansión permite la adición de otros periféricos*(joystick* impresoras...). Con la salida RS232 ya es posible la conexión directa a todos los periféricos que utilicen este sistema, especialmente las impresoras. Pero lo más destacable son los dos conectores situados a la altura del EAR y MIC del Spectrum, que hacen posible que esté conectado con otros Spectrums *(Local Area Network)*, hasta un total de 64 y una distancia máxima de 100 metros, pudiéndose intercambiar información o acceder a determinadas unidades de almacenamiento *(microdrives).*

A diferencia de los tradicionales sistemas de redes, ésta no necesita que todos sus componentes formen un círculo cerrado uniéndose el primero con el último. Es más, las conexiones de entrada del primero y de salida del último han de estar necesariamente libres. La única limitación estriba en que ningún ordenador integrado a la red puede desconectarse. Para poder utilizar la red, cada ordenador ha de tener un número de identificación, requiriéndose la interacción de los participantes (uno sólo no puede mandar nada si el otro no prepara la recepción). Es decir, que los participantes han de estar cara a cara. Para enviar un programa se utilizarían las instrucciones

FORMAT "n";1
SAVE * "n";2

El otro ha de preparar la entrada de la siguiente forma:

FORMAT "n";2
LOAD * "n";1

donde FORMAT indica el número de identificación y LOAD, la operación de carga del ordenador especificado.



COMO USAR EL MICRODRIVE

Este concepto de redes, de candente actualidad para los microordenadores, no parece ser de gran utilidad para el que nos ocupa. Suponemos que está dentro de la política de Sinclair de "dejarse sorprender", lanzando novedades al mercado y dejándole a éste que estudie su utilidad. Algunos ya ven en la red la posibilidad de utilizarla en las aulas, pero sus complicadas instrucciones de manejo no lo hace muy aconsejable.

INSTRUCCIONES DE MANEJO

Los comandos de referencia al *microdrive* están incluidas en el Spectrum. Algunas son las ya utilizadas con el *cassette* (LOAD, SAVE, VERIFY y MERGE), aunque con pequeños matices. Otras son totalmente nuevas (CAT, FORMAT, ERASE...). En general, no resulta nada sencilla la utilización de los nuevos comandos, si bien en muchos casos se puede agilizar su uso mediante rutinas en las que lo único que varíe sea el *microdrive* a utilizar y el nombre del archivo con el que se va a trabajar.

Veamos de forma resumida los nuevos comandos y la forma de utilizarlos, lo cual da una buena idea de las posibilidades reales del *microdrive*.

FORMAT "m";n; "Nombre". Formatea una cinta virgen para poder ser utilizada o una ya utilizada para su reutilización (borrando toda la información).

FORMAT "n";x. Identifica el Spectrum para su uso en la red.

FORMAT "t";x. Identifica la velocidad de transmisión de datos en la red: 50, 110, 300, 600, 1200, 2400, 4800, 9600, 19200.

SAVE *"m";n; "Nombre". Almacena un programa. Se puede añadir LINE XXXX para indicar, al igual que en el *cassette* la línea de comienzo.

VERIFY *"m";n; "Nombre". Verifica la perfecta grabación.

LOAD *"m";n; "Nombre". Carga el programa.

ERASE "m";n;"Nombre". Elimina un programa o archivo.

MERGE *"m";n; "Nombre". Mezcla un programa de cinta con el que se encuentra en memoria.

CAT n. Catálogo de programas y archivos contenidos en la cinta.

OPEN x,"m",n, "Nombre". Abre un archivo de datos, utilizando el *buffer* X (Cualquiera del 0 al 15).

PRINT # x. Grabación de datos en el archivo especificado por el *buffer* X.

INPUT # x. Lectura de datos del archivo especificado por el *buffer* X.

CLOSE # x. Cierra el archivo especificado por *buffer* X.

"m"	Indicativo de operación con *microdrive*.
n	*Microdrive* con que se trabaja (1 a 8).
x	*Buffer*.

INTERFACE 1. VARIABLES DEL SISTEMA

En el mapa de memoria del Spectrum ya estaba previsto el entonces futuro INTERFACE 1 y MICRODRIVES.

Los 58 bytes que ahora se incorporan al mapa de memoria permiten las siguientes funciones:

DECIMAL HEXADE NOMBRE FUNCION

23734 5CB6 FLAGS3 Bit 0. Activado utilizando elmodo extendido.
Bit 1. Activado con CLEAR #
Bit 2. Activado cuando se modifica ERRSP.
Bit 3. Activado con la Red Local.
Bit 4. Activado con LOAD.
Bit 5. Activado con SAVE.
Bit 6. Activado con MERGE.
Bit 7. Activado con VERIFY.

23735 5CB7 VECTOR Expansión intérprete, normalmente 01F0.

23737 5CB9 SBRT Acceso rutinas ROM 16 K de la forma:

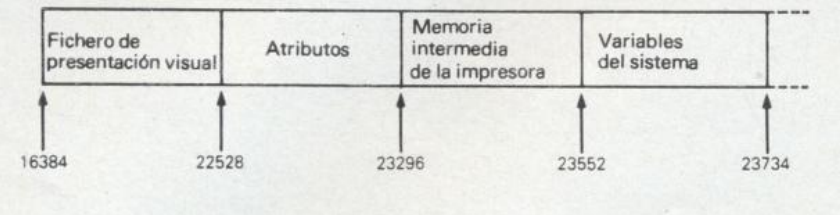
```
LD HL, valor
CALL
LD (5CBAH), HL
RET
```

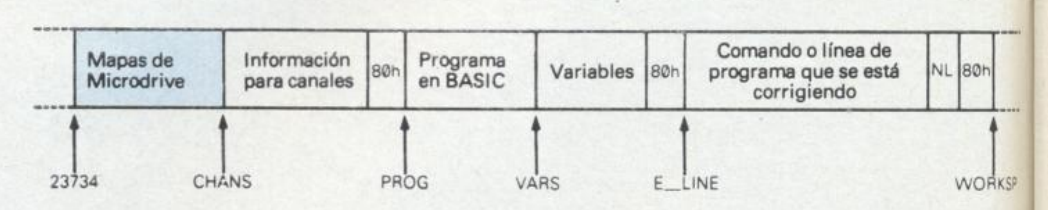
23747 5CC3 BAUD Velocidad transferencia de datos en baudios.
BAUD=(3500000/(26*código))—2

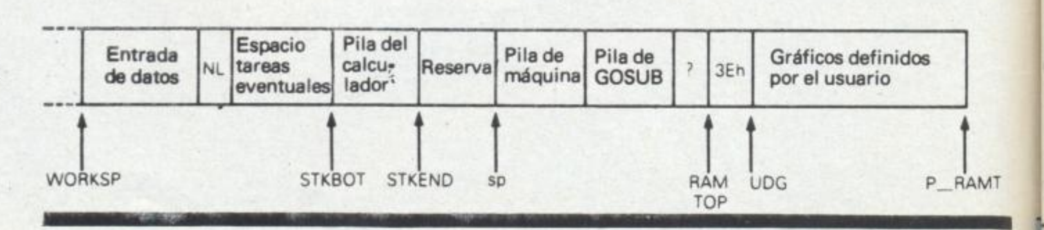
23749 5CC5 NTSTAT Código para la red local (1-64).

23750 5CC6 IOBORD Color de BORDER durante operaciones *input/output*.

23751 5CC7 SER--FL Control RS232. El primer byte es un *flag* y el segundo un carácter.







"Nombre" Nombre de la cinta, programa o archivo.

ERRORES DEL INTERFACE

Cuando salió al mercado el Spectrum ya iban incorporados los comandos a utilizar con el *Interface* 1. Tanta anticipación, necesaria si se tiene en cuenta la estructura de este ordenador, puede originar pequeños problemas en el uso de los comandos, pudiéndose encontrar con un funcionamiento anómalo.

Veamos cuáles pueden ser estos pequeños fallos:

1. ERRORES DE SINTAXIS

Hay una serie de comandos que en principio son aceptados, pero no obtendrá una respuesta adecuada al ordenador. Veamos estos comandos:

ERASE "nombre" Ej.: ERASE "PRUEBAS"

MOVE "nombre", "nombre". Ej.: MOVE "T1", "T2".

FORMAT "nombre". Ej.: FORMAT "PRUEBAS".

CAT. Ej.: CAT (sin número).

2. COMANDOS DE COLOR

Utilizar canales distintos del "K", "S", "P" puede afectar a los comandos de color con los que se trabaje posteriormente. El problema reside en las operaciones de *output* del *interface* que alteran el carácter permanente de los comandos de color.

3. COMANDO BREAK

Detener la ejecución de un programa no es tan sencillo con el *interface*. Si está trabajando con el *microdrive* o con la salida RS232, deberá presionar CAPS SHIFT y la barra espaciadora simultáneamente. Si opera con la red local y con la entrada RS232 bastará con presionar la tecla espaciadora.

4. OPERACIONES CON *CASSETTE*

Si se hace una incorrecta referencia a un archivo (por ejemplo, más de diez caracteres de longitud), en vez de aparecer el mensaje de error INVALID FILENAME se quedará sorprendido cuando aparezca en pantalla NONSENSE IN BASIC.

5. CODIGO MAQUINA

Aunque la mayor parte de la ROM es correcto hay dos pequeños fallos: READ--N no puede utilizarse al situarse el flag al final de la rutina y el código 2B es el mismo que 22, debido a una incorrecta entrada en la tabla de bifurcación (Dirección X19C9H).

6. LA SALIDA RS232

Si se utiliza el canal "t" para el listado de programas, se imprimirá un doble espacio entre las palabras clave. (THEN PRINT aparecerá como THEN PRINT).



Con unas dimensiones de 45 × 35 × 7 mm. el pequeño cartucho del microdrive contiene una banda continua de 15 m. cuya anchura es 1.9 mm. y puede llegar a almacenar entre 85 y 100 Kbytes.

23753 5CC9 SECTOR Control *Microdrive*. Sectores desde FFH a 04FBH.

23755 5CCB CHADD- Almacenamiento temporal para CHADD mientras se chequea la sintaxis de las líneas.

23757 5CCD NTRESP Código respuesta de la red local.

23758 5CCE NTDEST Inicio del *buffer* de la red, incluyéndose el código de Spectrum.

23759 5CCF NSTSRCE Código origen de la red local.

23760 5CDO NTNUMB Código bloque de la red local (0-65535).

23762 5CD2 NTTYPE Control red local (0=datos, 1=Fin de archivo).

23763 5CD3 NTLEN Longitud bloque de datos red (0-255).

23764 5CD4 NTDCS Chequeo bloque de datos red.

23765 5CD5 NTHCS Chequeo red (NTDEST-NTDCS).

23766 5CD6 D--STR1 Primer byte direccionamiento. Control *drive*, código *microdrive* (1-8), código red o código velocidad transmisión datos en baudios.

23768 5CD8 S--STR1 *Buffer* (0-15)

23769 5CD9 L--STR1 Tipo comunicación: 'M', 'N', 'T' o 'B'

23770 5CDA N--STR1 Longitud archivo.

23772 5CDC T--STR1 Comienzo archivo.

23774 5CDE D--STR2 Segundo byte direccionamiento. Comandos MOVE y LOAD.

23782 5CD6 HD--OO Direccionamiento comandos SAVE, LOAD VERIFY y MERGE.
Tipo datos: 0 — programa.
1 — tabla numérica.
2 — tabla alfanumérica.
3 — bytes.

23783 5CE7 HD--OB Longitud datos (0-65535)

23785 5CE9 HD--OD Comienzo datos. (0-65535)

23787 5CDB HD--OF Longitud programa. (0-65535)

23789 5CED HD--11 Línea de comienzo (ejecución automática).

23791 5CEF COPIES Número de copias realizadas con SAVE. 1 después de SAVE.

23792 Comienzo mapas *microdrive*.

7. EL COMANDO CLOSE #

Si se interrumpe un programa con el comando BREAK mientras está actuando el comando CLOSE#, no se recuperará la memoria utilizada por el *buffer*. Ello podría originar una importante acumulación de memoria sin utilizar.

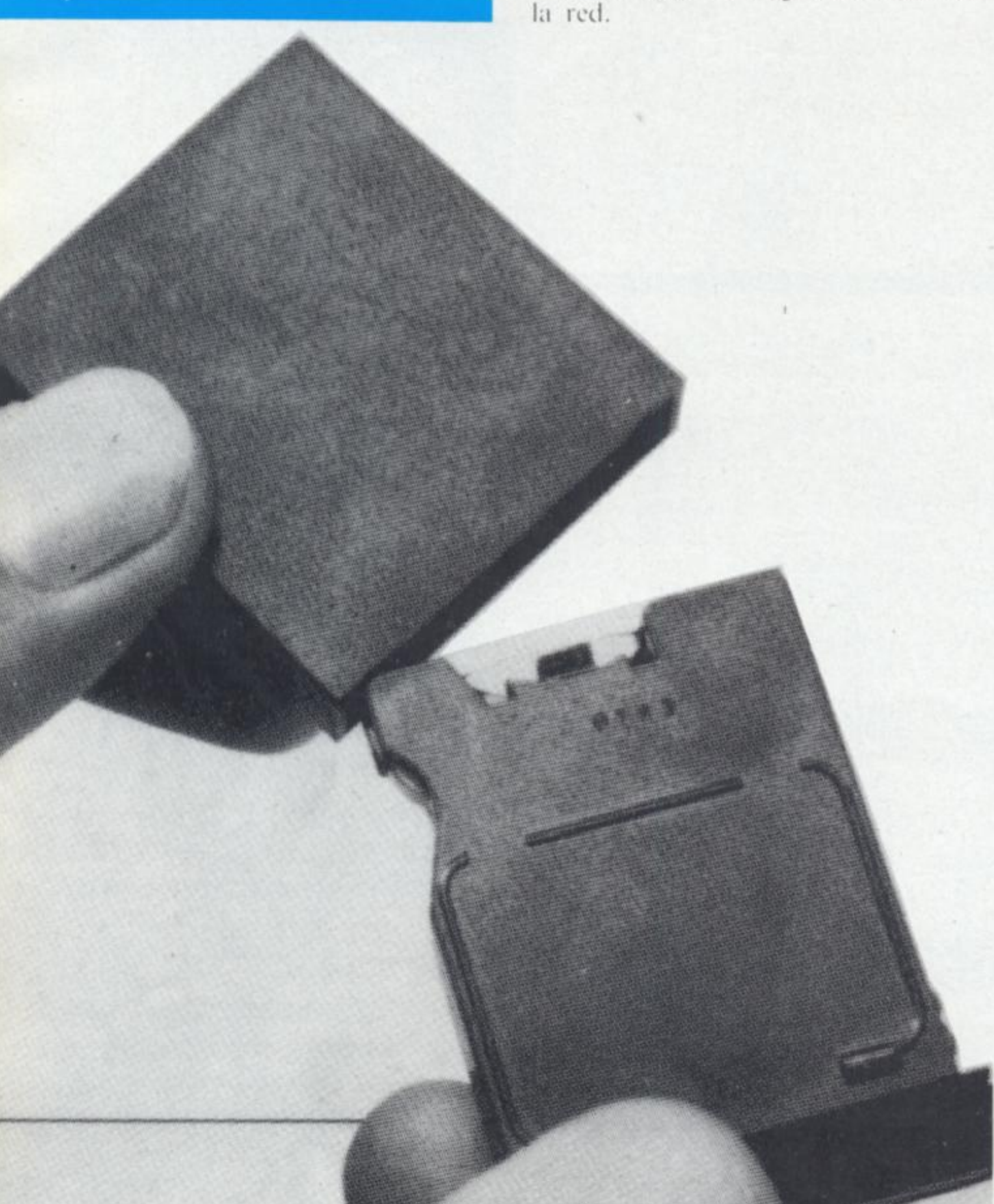
RED LOCAL O LAN (Local Area Network)

La posibilidad de conectar 64 Spectrums simultáneamente es una de las novedades más importantes, pero sobre todo más llamativas del *Interface* 1.

Cuando se trabaja con la red, se abre un *buffer* en el área designada como CHANS en el manual, nor-normalmente direccionado por el registro IX. La capacidad es de 276 bytes y su contenido es el siguiente:

0 Dirección 8.
2 Dirección 9.
4 'N'.
5 Dirección de la subrutina de salida en ROM.
7 Dirección de la subrutina de entrada en ROM.
9 Dirección 276.
11 NCIRIS Código de destino en la red.
12 NCSELF Código del Spectrum en la red.
13 NCNUMB Código de bloque.
15 CTYPE Tipo de datos.
16 NCOBL Número de bytes del bloque de datos.
17 NCDCS Chequeo datos.
18 NCHCS Chequeo datos.
19 NCBL Dirección último carácter en el *buffer*.
20 NCB Número de bytes del *buffer*.
21 *Buffer* de 225 bytes.

COMO USAR EL MICRODRIVE



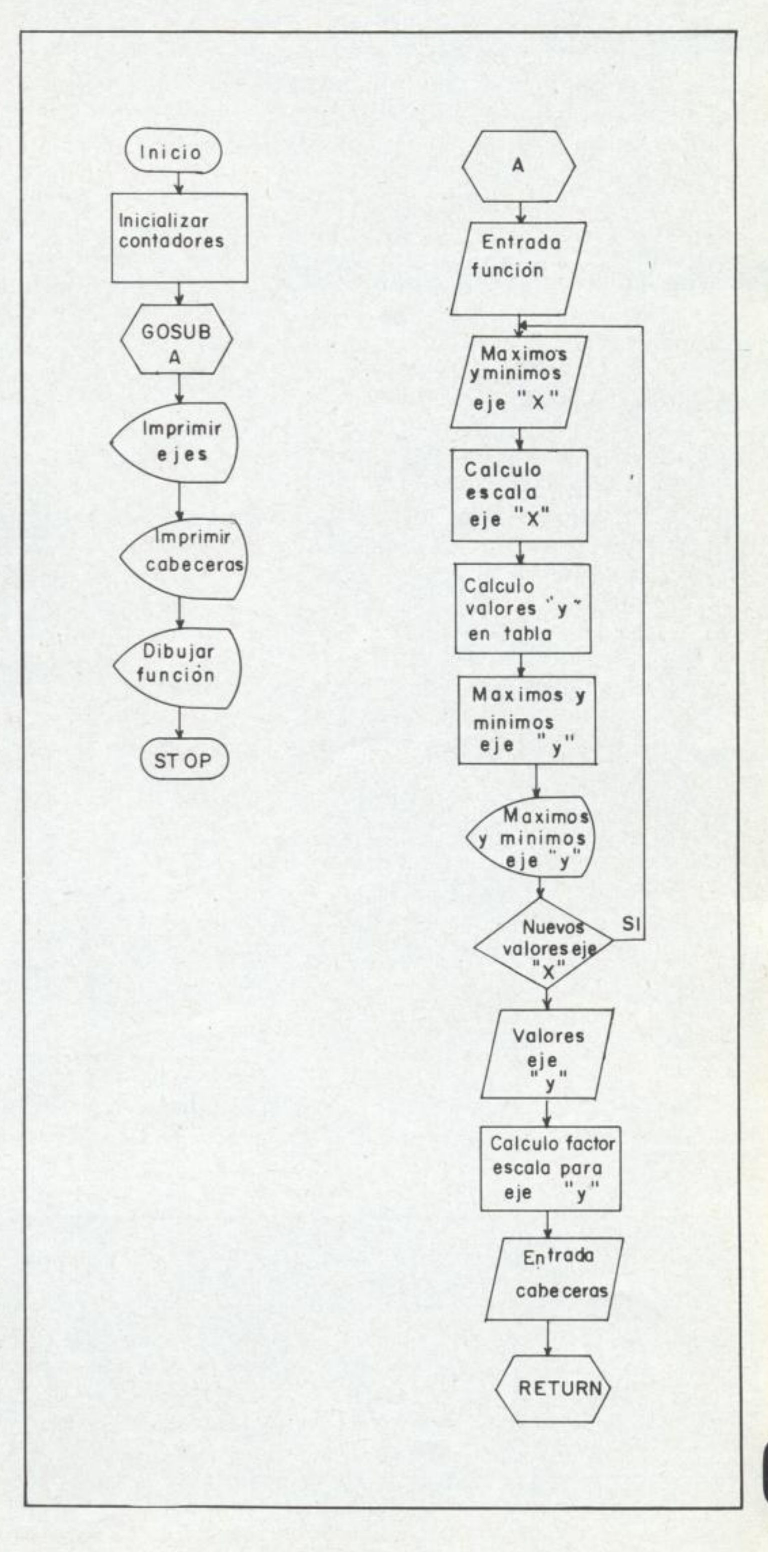
RESUMIENDO

Bajo el nombre de *microdrive* puede disponer de un sistema de almacenamiento intermedio en prestaciones y calidad a las cintas de cassette por un lado y los discos por otro. Además con la obligada colaboración del Interface I nuestro ordenador se hace más interesante. Pudiéndose conectar hasta 64 Spectrums, el Interface I nos permite acceder a cualquier tipo de información que sea tratada por cualquier ordenador conectado a la red. Esta es sólo una de las múltiples ventajas del Interface, ya que seguidamente tenemos una conexión RS232 que permite conectar periféricos que utilicen este tipo de conexión especialmente impresoras de más calidad que la pequeña ZX Printer. Finalmente, a la izquierda del Interface tenemos el *port* que controla los microdrives, con la particularidad de que nos permite tener acceso hasta 8 *microdrives* con una capacidad total de hasta 680k, lo que nos permite convertir a nuestro Spectrum en algo más que un ordenador personal.

Las ventajas respecto al clásico *cassette* son claras y permiten ya hablar de un sistema de almacenamiento rápido, flexible, suficiente y de bajo precio para este ordenador. Como contrapartida, nos encontramos con instrucciones de manejo excesivamente complicadas, un desgaste excesivo de la cabeza de lectura/escritura debido a la acción abrasiva de la cinta y un precio elevado del cartucho y unas prestaciones muy inferiores a las de una unidad de discos convencional.

PROGRAMACION A FONDO





En esta sección nos proponemos cada mes analizar un programa en profundidad. Partiendo siempre de un problema determinado, se analizará la forma en que puede resolverse por el ordenador. Junto al listado del programa se incluirá el organigrama y su representación en pantalla.

En esta ocasión vamos a "profundizar" en el análisis de la representación gráfica de funciones. Un tema que esperamos sea del agrado de muchos, al poder llegar a representar cualquier función.

Problema

El objetivo es construir un programa que realice la representación gráfica de varias funciones y=f(x), en el primer cuadrante (X e Y positivas) mostrando títulos y puntos de los ejes.

Distintas funciones requieren un INPUT o una línea que se cambie para cada función. La representación de títulos implica definir una área de graficación inferior al tamaño de la pantalla. Luego tendremos tres partes diferenciadas:

1. Introducir una función.
2. Introducir las variaciones de X e Y.
3. Dibujar la gráfica.

Los dos primeros puntos son relativamente sencillos. El trazado de la gráfica y los valores de los ejes de coordenadas es lo más complicado. Para trazar la gráfica podemos partir de una posición determinada (por ejemplo, pixels X=40, Y=52). Obteniendo los valores de Y y almacenándolos en una tabla, puede pasarse a la pantalla mediante un factor de escala: DY=120/(Y2—Y1), siendo Y2 e Y1 los valores máximos y mínimos del eje Y, respectivamente. La posición en el eje X será (N(F)—Y1) *DY + 52, siendo N(F) el elemento f-esimo de la tabla N correspondiente a un valor del eje Y.

Se puede dividir en cuatro partes el eje Y, y en seis el eje X. Otras combinaciones son posibles, pero así se logra dividir la zona de graficación en cuadrados, haciéndose fácilmente visible la evolución de las curvas para los distintos valores introducidos.

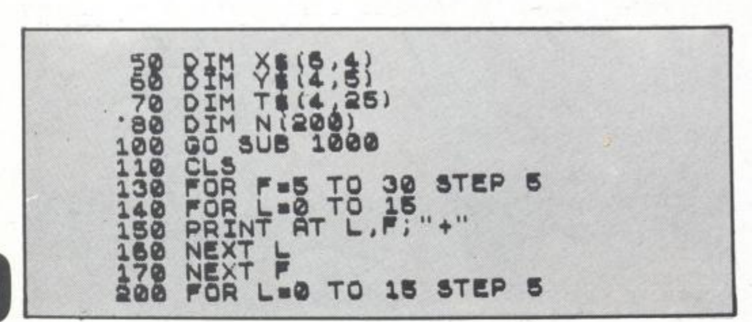
Para evitar complicaciones innecesarias se puede introducir los valores del eje y manualmente. Para ello, el ordenador debe indicar los valores mínimo y máximo. Así, si estos valores fuesen 0.12 y 3.6, se podría elegir para el eje Y los valores 0, 1.5, 3.0 y 4.5.

Tabla de variables

X$(6,4) Array valores eje X.
Y$(4,5) Array valores eje Y.
T$(4,25) Array títulos ejes y rótulo gráfico.
N(50) Array valores calculados de Y.
F Variable de bucle.
L Variable de bucle.
F$ Función.
X Valor de X.
MINX Valor mínimo de X.
MX Valor máximo de X.
DX Incrementos de X.
Y Valores de Y.
MAXY Valor máximo de Y.
MINY Valor mínimo de Y.
M$ Opción elegida del menú.
Y1 Valor mínimo del eje Y.
Y2 Valor máximo del eje Y.
DY Factor de escala para graficación.

Resultado

Para probar la validez del programa se puede empezar por una función conocida, como la función y=seno X+X. DIBUJO

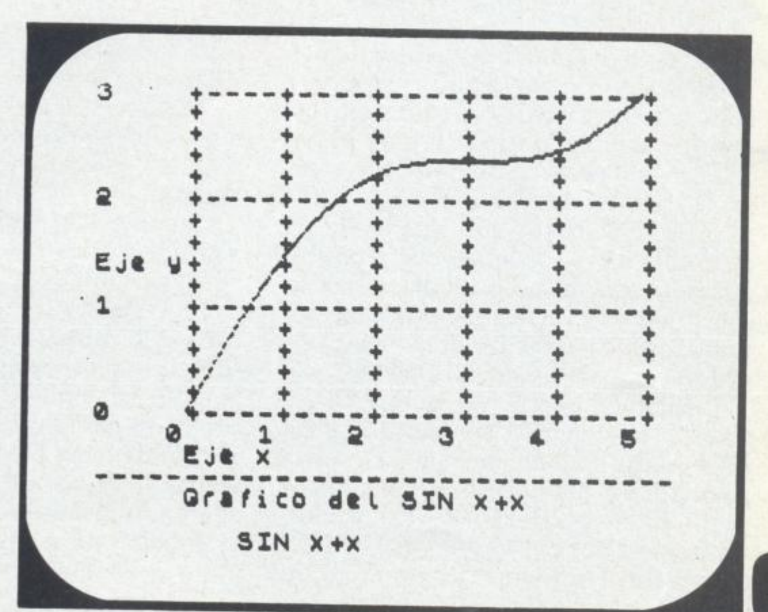


```
 50 DIM X$(6,4)
 60 DIM Y$(4,5)
 70 DIM T$(4,25)
 80 DIM N(200)
100 GO SUB 1000
110 CLS
130 FOR F=5 TO 30 STEP 5
140 FOR L=0 TO 15
150 PRINT AT L,F;"+"
160 NEXT L
170 NEXT F
200 FOR L=0 TO 15 STEP 5
```

PRINCIPIO
SUB. NIVEL Y VARIANTE
SUB. GRAFICOS DEFI.
SUB. DIBUJO LABER
APERTURA de ACCESOS
BUCLE PRINCIPAL
VARIABLES RELOJ
SUB INKEY $
MOVIMI. CORRECTO? — SI — INCORRECT
NO
DIBUJO de CAMINO
CONTROL DE BARRAS LIMITE ?
LIMITE? — SI — HA EXCEDIDO EL LIMITE DE MOVIMIENTOS
NO
DIBUJO BARRA
SITUACION DENTRO DEL LABERINTO
* — SI — SALIDA DEL LABERINTO — OTRA?
NO
MOVIMIENTO DE LAS AGUJAS DEL RELOJ
S/N
NO
FIN
7100 GRABACION

PROGRAMACION A FONDO

```
 205 FOR F=6 TO 29
 206 IF F/5=INT (F/5) THEN NEXT
F
 210 PRINT AT L,F;"-"
 220 NEXT F
 230 NEXT L
 250 FOR F=1 TO 6
 260 PRINT AT 16,(F*5-1);X$(F)
 270 NEXT F
 280 FOR F=1 TO 4
 290 PRINT AT 20-F*5,0;Y$(F)
 300 NEXT F
 310 PRINT AT 17,5;T$(3)
 320 PRINT AT 18,0;"-------------
-----------------------"
 330 PRINT AT 19,5;T$(1);AT 21,8
;T$(2)(1 TO 20)
 340 PRINT AT 8,0;T$(4)(1 TO 5)
 360 FOR F=1 TO 200
 370 PLOT F+40,(N(F)-Y1)*DY+52
 380 NEXT F
 390 GO TO 5000
1000 PRINT "GRAFICO DE UNA FUNCI
ON"
1010 PRINT ,,"RUTINA DE CALCULO"
1020 PRINT "TECLEA LA FUNCION QU
E QUIERES    REPRESENTAR DE LA FO
RMA:         Y=FUNCION X."
1030 PRINT "TECLEA LA FUNCION CO
MO UNA CADE-NA. PUEDE HABER ALGU
N ERROR CO- MO RESULTADO DE EVAL
UAR ALGUNAS FUNCIONES. SI ESTO O
CURRE, SUS- TITUYA LA CADENA."
1050 INPUT F$
1060 CLS
1070 PRINT ,,"VALOR MINIMO DE X?
"
1080 INPUT MINX
1090 PRINT ,,"VALOR MAXIMO DE X?
"
1100 INPUT MX
1120 LET DX=200/(MX-MINX)
1140 LET X=MINX
1150 FOR F=1 TO 200
1160 LET Y=VAL F$
1170 LET N(F)=Y
1180 LET X=X+(1/DX)
1190 NEXT F
1210 LET MAXY=N(1)
1220 LET MINY=N(1)
1230 FOR F=2 TO 200
1240 IF N(F)<MINY THEN LET MINY=
N(F)
1250 IF MAXY<N(F) THEN LET MAXY=
N(F)
1260 NEXT F
1270 PRINT "EL VALOR MINIMO DE Y
 ES:",MINY
1280 PRINT "EL VALOR MAXIMO DE Y
 ES:",MAXY
1290 PRINT ,,"ELIGE LOS VALORES
ADECUADOS PA- RA Y, EJE (Y) O RE
ESTABLECER LOSVALORES DE X (X)?
TECLEA X o Y."
1300 INPUT M$
1310 CLS
1320 IF M$="X" THEN GO TO 1070
1340 PRINT "TECLEA EL VALOR DEL
MINIMO DE Y,(<=";MINY;") PARA RE
ALIZAR LA"'"ESCALA."
1350 INPUT Y1
1360 PRINT "TECLEA EL VALOR DEL
MAXIMO DE Y,(<=";MAXY;") PARA RE
ALIZAR LA"'"ESCALA."
1370 INPUT Y2
1380 LET DY=120/(Y2-Y1)
1390 PRINT "EL EJE Y TIENE 3 DIV
ISIONES Y    120 PUNTOS (PLOT)."
1400 PRINT "TECLEA LOS 4 VALORES
 DE LA ESCA-LA Y. MAX 5 CARACT."
1410 PRINT "PULSA ENTER PARA ESP
ACIO BLANCO"
1420 PRINT "MINIMO(1)?"
1430 INPUT Y$(1)
1440 FOR F=2 TO 4
1450 PRINT "VALOR ";F;"?"
1460 INPUT Y$(F)
1470 NEXT F
1480 CLS
1490 PRINT "TECLEA LA ESCALA DEL
 EJE X"
1500 PRINT "MINIMO DE X ES ";MIN
X
1510 PRINT "MAXIMO DE X ES ";MX
1520 PRINT ,,"HAY 5 DIVISIONES Y
 200 PUNTOS   (PLOTS) EN EL EJE
X"
1530 PRINT "VALOR DE LA ESCALA M
AX. 4 CARAC.PRIMERO TECLEA EL VA
LOR MINIMO."
1540 FOR F=1 TO 6
1550 PRINT "VALOR DE ESCALA ";F;
"?"
1560 INPUT X$(F)
1570 NEXT F
1590 CLS
1600 PRINT "TECLEA EL TITULO DEL
 GRAFICO     (COMO MAXIMO 25 CARA
C.)"
1610 INPUT T$(1)
1620 PRINT "TECLEA LA FUNCION O
UN SUBTITULO(MAXIMO 20 CARACT.)"
1630 INPUT T$(2)
1640 PRINT "TECLEA EL NOMBRE DEL
 EJE X.(MAX 20 CARACTERES.)"
1650 INPUT T$(3)
1660 PRINT "TECLEA EL NOMBRE DEL
 EJE Y.(MAX 25 CARACTERES.)"
1670 INPUT T$(4)
1680 RETURN
5000 STOP
```



Todavía hay muchos para los que el Spectrum es un sofisticiado juguete del desarrollo tecnológico. Sofisticado, pero juguete al fin y al cabo.

Nanoordenador, ordenador doméstico o *home computer*, según la terminología anglosajona, son las diversas denominaciones usadas para calificar a este ordenador, convertido en el regalo de fin de curso más preciado por las nuevas generaciones.

Hasta el momento, hemos asistido a una enorme proliferación de ordenadores, evidenciándose una enorme laguna: el ordenador personal para aplicaciones "profesionales". Sin duda el nuevo QL de Sinclair pretende llenar ese campo. Mayores capacidades, disco duro, BASIC más potente... a un precio igualmente intermedio.

Pero quienes tienen un ordenador desde hace tiempo ya saben que el problema no está sólo en la compra de uno nuevo, sino en la pérdida del *software* del viejo. Es por ello que vale la pena resistir e intentar ampliar al máximo las capacidades del ordenador que, dicho sea de paso, rara vez entra dentro de la política de comercialización de las casas proveedoras.

El *microdrive* constituye hasta el momento la ampliación más importante al Spectrum, sobre todo la más conocida en nuestro país, pero no es la única. Desconocido en el mercado español, aparecieron en marzo del 83 una serie de expansiones modulares bajo el nombre de Basicare. A pesar de sus enormes ventajas desconocemos si llegará a comercializarse en España. Veamos sus características más notables, objeto de este artículo.



Basicare es una serie de módulos conectados entre sí por la parte superior e inferior de cada uno de ellos. Existen diversos módulos con diferentes posibilidades, de forma que el usuario construye —podríamos decir que edifica— su propio ordenador de acuerdo con sus necesidades. Gracias a este acople vertical se consigue un gran aprovechamiento de espacio y se evitan las conexiones por cable.

Inicialmente se pensó en el ZX81, a fin de potenciarle sus puntos más débiles (principalmente la escasa capacidad de memoria). Pero el proyecto era más ambicioso y ofreciendo un procesador de 16 bit y el MSDOS (Microsoft Disk Operation System) se abría la puerta del abundante *software* del IBM/PC y sus clónicos. El ZX81 fue sólo el principio.

En junio del presente año ya estaba disponible para el Spectrum, llenando el vacío existente para aplicaciones "serias".

Ocho son los módulos que se pueden conectar:

1. PERSONA

El solo no hace nada, pero tampoco harían nada todos los demás sin él, pues su misión es establecer el bus de conexión para los demás módulos. En el caso de tener el Spectrum de 48K es necesario hacer un pequeño cambio, anulando las últimas 32k (si por error lo conecta directamente, es probable que no le ocurra ningún daño a su ordenador, pero no se lo aconsejamos), por lo que resulta menos ventajoso para este equipo.

Con este módulo se pueden controlar hasta cinco módulos adicionales. Para expansiones superiores a cinco módulos existe una expansión del propio módulo PERSONA.

2. RAM

Existen tres módulos de ocho, 16 y 64K de memoria. Con ellos puede llegar hasta cuatro megabytes de memoria (1Mg=1.000k). Lógicamente, la configuración del Z80 no permite estos lujos directamente. Para ello ha de configurarse un módulo llamado minimapa, que organiza la memoria incorporada en tres grupos de memoria y cada grupo en segmentos, a fin de permitir su manipulación.

Basicare: para convertir su Spectrum en un auténtico Ordenador Profesional

En el primer grupo reside la ROM, que ha de estar presente para el procesador de forma constante. A los otros dos grupos de memoria se accede según las necesidades. El se-

Módulos de expansión de memoria de 8, 16 y 64K, pudiéndose llegar a 4 megabytes.

gundo grupo contiene segmentos de 32K donde residen los programas en BASIC. El tercer grupo tiene segmentos de 16K para ubicar programas en código máquina o rutinas en BASIC.

La figura 1 muestra el mapa de memoria y la interacción de los distintos grupos.

La expansión de memoria tiene, por tanto, unas grandes limitaciones, ya que sólo trabaja con el límite de la RAMTOP, 32768 o direcciones inferiores. Es decir, que programas de más de 16K no pueden utilizarse si está conectada la Basicare. La ampliación de memoria resulta especialmente interesante para la utilización de otros módulos, como el DROM o para almacenamiento diverso, como puede ser diversas pantallas en cualquiera de sus zonas de memoria de 16K, pero no para realizar programas más complejos, sino todo lo contrario: deberá desconectarse la Basicare para utilizar programas de más de 16K.

3. PERICON
Este módulo dispone de tres puertas de 8 bits para operaciones de entrada de datos, presentando diversas modalidades. Así PERICON A permite potenciar la comunicación vía *ports*, convertidas en simples direcciones de memoria accesibles por las instrucciones PEEK y POKE. PERICON B amplía el número de puertas. PERICON C es un interface Centronics con el que poder utilizar impresoras con las clásicas instrucciones LLIST, LPRINT, para campos de impresión de hasta 80 columnas y 66 líneas por página.

4. DROM
Uno de los más importantes módulos, con el que dispondrá de 8K de memoria permanente ROM para rutinas propias. Rutinas de impresión, *utilities*, etc., todos accesibles inmediatamente.

Junto con el módulo se incluyen unas rutinas interesantes: auto enumeración de las líneas del lenguaje BASIC, un comando RESET en código máquina, y una rutina de lectura del módulo del reloj (CLOCK) que asigna los valores relativos a la fecha y tiempo, en la variable t$ (aunque ello presenta el inconveniente de su eliminación de memoria con cada RUN o CLEAR que se efectúa, necesitando hacer una posterior llamada al módulo del reloj).

5. TOOLKIT
Junto con el anterior, es uno de los módulos más interesantes. A las ruti-

Diseño modular según sus necesidades.

nas EPROM creadas se puede acceder directamente desde programa si se las guarda en este módulo que, al igual que el DROM, puede tener 8K. Como complemento al anterior es especialmente útil, ya que se pueden desarrollar y probar rutinas en DROM y después pasarlas al TOOLKIT, una vez verificado su correcto funcionamiento.

6. LINK A
La conversión analógico-digital, tan de moda para el control de

aparatos domésticos, tiene aquí su módulo. La lectura se realiza mediante las instrucciones PEEK y POKE.

Cada módulo tiene ocho canales de comunicación, pudiendo conectarse un máximo de cuatro módulos. El voltaje debe estar comprendido entre cero y cinco positivo, pudiendo resultar dañado el ordenador si se aplican corrientes ñegativas o superiores a cinco voltios (¡cuidado con su estéreo!).

7. CLOCK

Se trata de un auténtico reloj con batería que ofrece los datos de tiempo y día de la semana, mes y año, lectura y puesta en hora del reloj. Junto al módulo se suministra un *cassette* con dos programas para la lectura y puesta en hora del reloj. La batería permite una autonomía de tres meses y se carga automáticamente.



8. SONUS

Módulo sintetizador de voz y generador de sonido, cubriéndose el vacío sonoro impuesto por Sinclair.

Según cuáles sean sus necesidades, puede "montarse" su propio ordenador a la medida. Pero no se haga demasiadas ilusiones: con estos módulos puede tener unas potentes herramientas pero sólo eso. Todo el trabajo lo tendrá que hacer usted, ya que no existe *software* por el momento a excepción de lo que hemos comentado suministrado con los módulos. El *hardware* está ahí, el *software* requiere de profesionales que se aventuren con el Spectrum. Un solo ejemplo: basta un par de módulos para tener 64K, pero ¿ha visto en el mercado algún programa que necesite más de 48K? E incluso si se trata de un programa que necesita más de 16K no será factible por la técnica de "paginación" de la memoria en bloques de 16K, como hemos visto. Todo está por hacer, incluso por rehacer, como ocurre con los manuales donde la diferencia en el uso de valores decimales-hexadecimales dista mucho de estar clara o la explicación para su utilización por el ordenador, no distinguiéndose en muchos casos entre el ZX81 o el Spectrum.

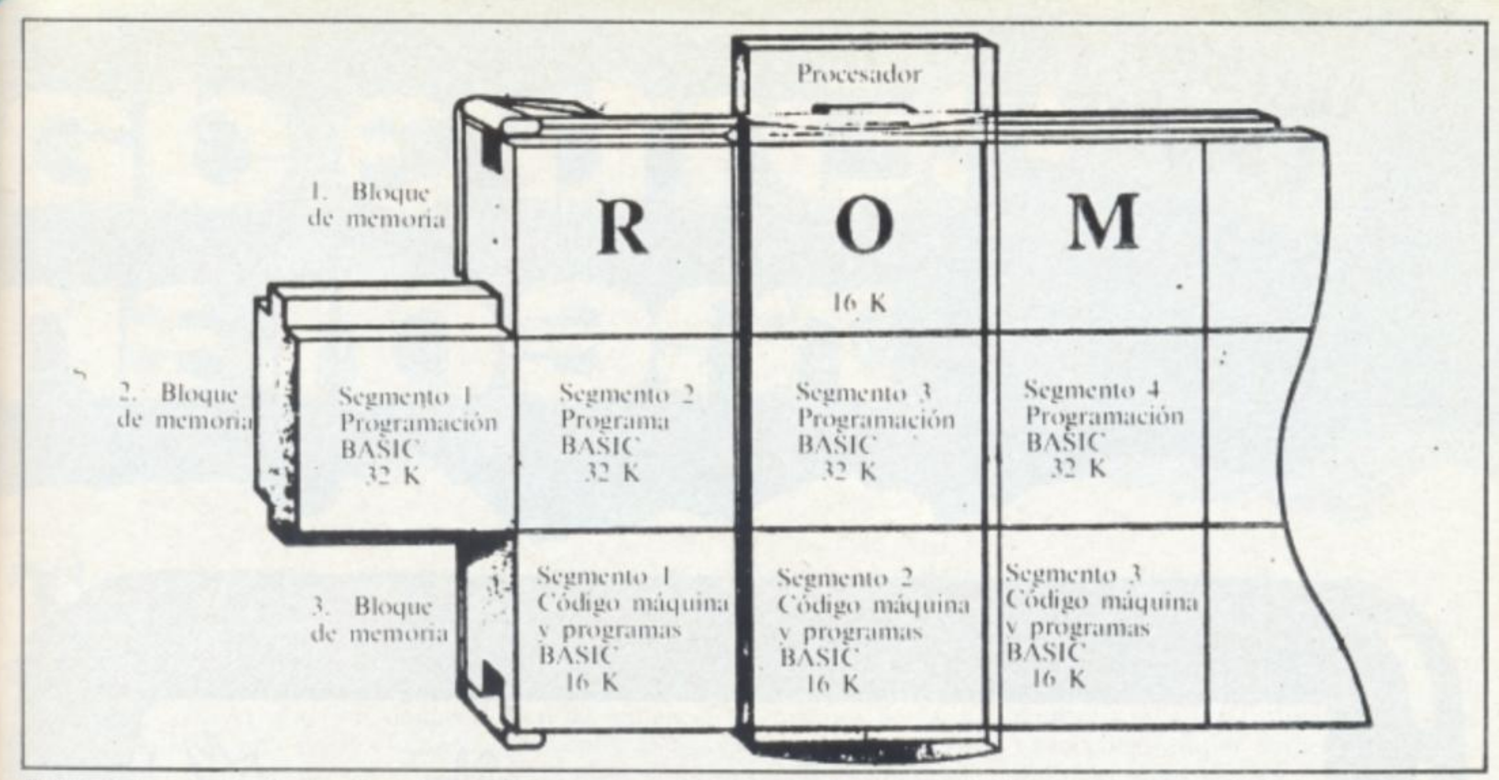
El hardware ya está hecho. El software es cosa suya.

Otro de los problemas a tener en cuenta es que el módulo PERSONA no tiene el *port* de expansión, por lo que no podrá incorporar otros periféricos tales como *joystick* o el *interface* 2, pero sí el *interface* 1, con lo que podrá seguir trabajando con el *microdrive*, buen compañero de Basicare, al añadirse la posibilidad de utilizar la Network y la salida RS232.

En cualquier caso, una vez más la técnica dio el paso hacia adelante y ahora hay que alcanzarla. Valore sus necesidades y evalúe las ventajas e inconvenientes por si le es de interés.

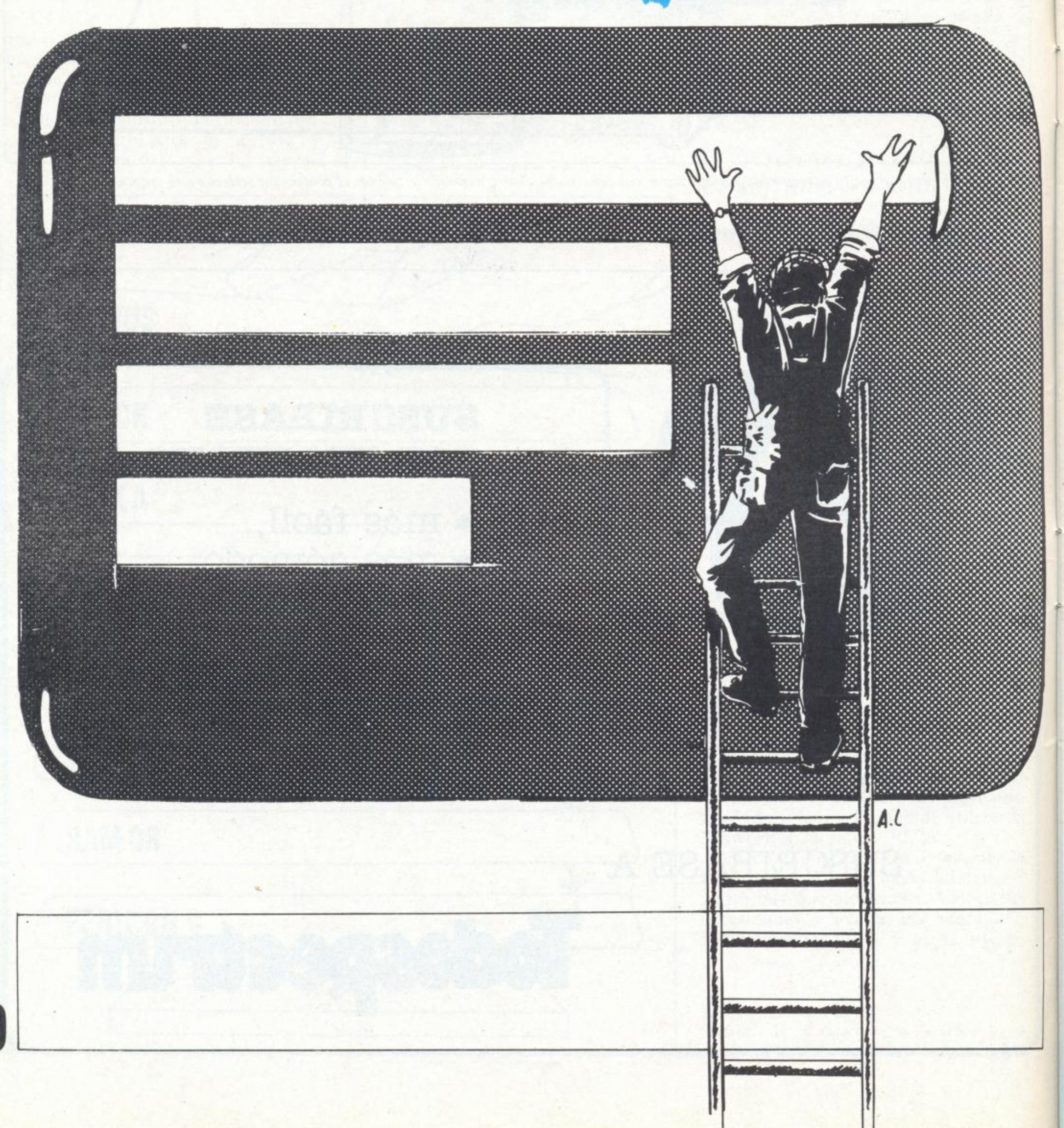
SONUS
CLOCK
LINK A
TOOLKIT
DROM
PERICON A
RAM 08
PERICON C

Basicare: usted decide la "altura" de su Spectrum de acuerdo a sus necesidades.



Expansión de memoria. El direccionamiento se logra con una técnica de "paginación", donde el procesador selecciona la página a leer.

Entrada mediant



de datos
te máscaras

Todo programa requiere generalmente de entrada de datos por pantalla. Para ello el Spectrum y el BASIC en general dispone de la instrucción INPUT con la que, sin demasiadas complicaciones puede lograr su objetivo. Pero si es usted un perfeccionista, seguro que más de una vez ha pensado en la posibilidad de confeccionar una ficha de entrada de datos y realizar las anotaciones en los campos al efecto diseñados en la ficha, es decir, en utilizar la técnica de "máscaras". Todo ello, lógicamente, a través de la pantalla de su televisor.

Para lograr esto la instrucción INPUT no parece ayudar mucho, pues siempre efectúa la entrada de datos en la última línea de la pantalla. Por el contrario, la instrucción PRINT AT resuelve el problema a medias, ya que puede imprimir pero no leer datos. Pero no hemos de desesperarnos tan pronto, y confiar en la potente instrucción INKEY$. Con ella se pueden leer datos sin que aparezcan en pantalla y con la instrucción PRINT AT se puede representar los caracteres leídos allí donde queramos.

Dicho así puede resultar demasiado trivial y frustrante para el perfeccionista que a buen seguro no se conformará con utilizar el INPUT. Para realizar una auténtica máscara han de indicarse los campos de cada concepto: la posición donde empieza y termina o, lo que es lo mismo, las coordenadas iniciales y finales de X y la coordenada de Y.

También podemos indicar el tipo de datos del campo, pudiendo ser alfanumérico o numérico y, en este último caso, real o entero, con lo que evitar errores en la introducción de datos con oportunos mensajes o tonos de error.

En el programa adjunto se ilustra este procedimiento en la rutina que va de la línea 9000 a 9095. (Una numeración alta permite partir de las rutinas construidas para posteriormente realizar el programa sin tener que repetir dichas rutinas). Las líneas 10 a 5113 únicamente sirven de ejemplo de utilización de la rutina, que veremos a continuación en detalle.

Las variables X9, T9, Y9 representan las coordenadas del campo y K9 el tipo de dato. Para controlar la posición dentro de un campo se utiliza la variable X8 que parte del valor de X9 y que podrá alcanzar como máximo el valor de T9, es decir, los valores de columna inicial y final, respectivamente. También sirve para retroceder (anular caracteres) sin rebasar el valor de X9.

Entrando por la línea 9026 inicializamos los valores de X8 y X$, que será la variable que contenga la información del campo T9-X9+1.

La línea 9030 nos informa con el signo interrogación de la posición en que se introducen los datos, por efecto de la instrucción INKEY$ de la siguiente línea. Una vez pulsada una tecla se comprueba varias cosas:

1. Si se ha pulsado ENTER. El código ASCII de esta tecla es el 13. La instrucción 9035 lo reconocería como fin de entrada de datos para ese campo.
2. Si se ha pulsado la tecla CAPS SHIFT + 5: el retroceso, a fin de borrar caracteres escritos. Dicha tecla responde al código ASCII 8, con lo que se pasaría a la línea 9050, donde se comprueba que se puede retroceder (si x8≠x9) y se imprime un blanco

```
  10 REM ************************
*          * Utilizacion mascaras
*          * con el Spctrum.
*          ************************
*
  15 DIM t$(99,22): DIM a$(99,22
): DIM e$(99,22): LET n=0: LET y
$=""
  20 CLS : PRINT ''"1. Altas"'"2
. Bajas"'"3. Cambios"
  30 LET q$=INKEY$: IF LEN q$=0
THEN GO TO 30
  40 GO TO 100*VAL (q$)
  95 REM -----------
 100 REM     Altas
 101 REM -----------
 110 GO SUB 5000: LET t=1: GO SU
B 5070: LET n=n+1: GO TO 20
 199 REM -----------
 200 REM     Bajas
 201 REM -----------
 210 GO SUB 5000: LET t=2: GO SU
B 5070: LET n=n-1
 220 LET t$(c)="": LET a$(c)="":
 LET e$(c)="": GO TO 20
 299 REM -----------
 300 REM    Cambios
 301 REM -----------
 310 GO SUB 5000: LET t=0: GO SU
B 5070: GO TO 20

5000 REM ***********************
         * Impresion mascara *
         ***********************

5010 CLS : PRINT AT 0,12;"Biblio
teca"
5020 PLOT 95,167: DRAW 80,0
5030 PRINT AT 2,2;"Codigo "; INK
 4;"███"
5040 PRINT AT 3,2;"Titulo "; INK
 4;"██████████████████████"
5050 PRINT AT 4,2;"Autor  "; INK
 4;"██████████████████████"
5060 PRINT AT 5,2;"Editor "; INK
```

Entrada de datos mediante máscaras

(carácter 32 en ASCII) para borrar la información.

3. Si el tipo de dato es el correcto de acuerdo al valor de K9.

— Si es un dato real (K9=1), por efecto de la línea 9060 sólo se admite la entrada de números y el signo menos (—).

— Si el dato es entero (K9=2) sólo se admiten losnúmeros (línea 90-65).

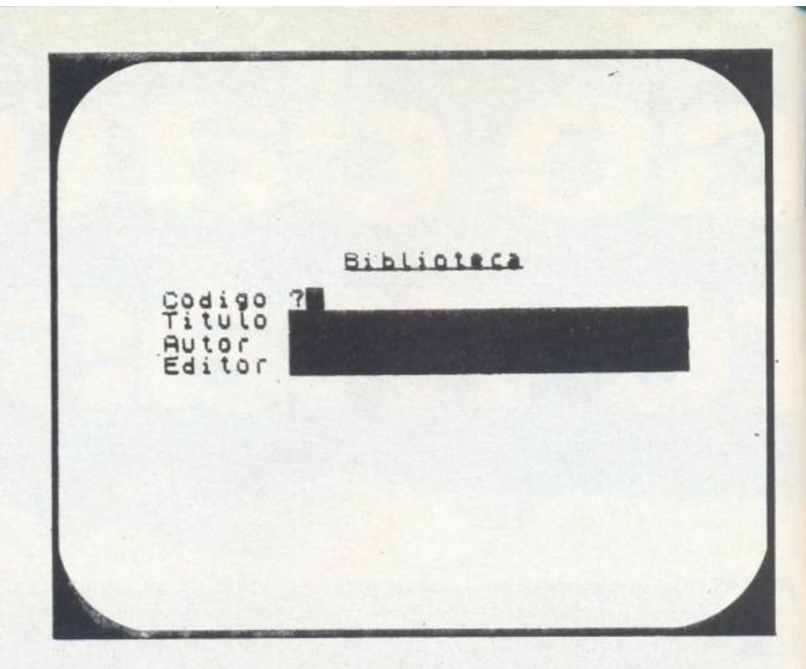
— Y si se trata de datos alfanuméricos (K9=3) son válidos todos los caracteres comprendidos entre el código 32 y 164 ASCII (Línea 9070).

Un dato incorrecto se imprime (línea 9084) y se pide un nuevo carácter, siempre que no se haya llegado al final del campo (x8=t9), en cuyo caso se asigna el valor del campo a la variable x$ de forma automática sin necesidad de presionar ENTER. La rutina devuelve el valor X para datos numéricos (k9=1 ó K9=2) y x$ para los alfanuméricos (K9=3).

Decíamos que entrábamos por la línea 9026 en vez de hacerlo por la 9000. Así se evita el paso por la 9015 y 9020, cuyo objeto no es la introducción de datos, sino la impresión de datos conocidos para los distintos campos.

La utilización de esta rutina puede parecer algo engorroso al principio. Imaginemos que se desea guardar referencias de libros (código, título, autor y editor) tal y como se haría en cualquier biblioteca (el código podría servir para clasificar por materias). Las líneas 5000 a 5060 realizan la máscara, que nos servirá para todas las operaciones de actualización, es decir, para dar de alta nuevos libros, anularlos o efectuar cambios en sus datos. De idéntica forma las instrucciones 5070 a 5113 permiten introducir datos en la máscara para su actualización. El control de los distintos tipos de actualización se realiza mediante la variable t (=1 par altas; =2 para bajas; =0 para cambios). Así, si el valor de "t" = 1 se pueden introducir los datos en los distintos campos; si t=2 sólo se pide el código para darlo de baja, y si t=0 previamente se coloca en el campo su valor para después poder realizar los cambios oportunos.

La ficha que se muestra es sólo un sencillo ejemplo.




```
 4;"[redacted]": RET
URN

5070 REM *****************
         * Entrada datos*
         *****************

5080 LET x9=9: LET t9=10: LET y9
=2: LET k9=2: GO SUB 9026: LET c
=x: IF t=2 THEN RETURN
5090 LET t9=30: LET y9=3: LET k9
=3
5091 IF t THEN GO SUB 9026: GO T
O 5093
5092 LET x$=t$(c): GO SUB 9020
5093 LET t$(c)=x$
5100 LET y9=4
5101 IF t THEN GO SUB 9026: GO T
O 5103
5102 LET x$=a$(c): GO SUB 9020
5103 LET a$(c)=x$
5110 LET y9=5
5111 IF t THEN GO SUB 9026: GO T
O 5113
5112 LET x$=e$(c): GO SUB 9020
5113 LET e$(c)=x$: RETURN

9000 REM ***********************
         * Rutina entrada da- *
         * tos por mascara.   *
         *                    *
         * x9=columna inicial.*
         * t9=columna final.  *
         * y9=fila.           *
         * k9=tipo de dato    *
         *   1-Real 2-Entero  *
         *     3-Alfa.        *
         ***********************

9015 PRINT AT y9,x9; INK 4;x: GO
 TO 9025
9020 PRINT AT y9,x9; INK 4;x$
9025 LET x8=x9: PRINT AT y9,x8;x
$
9026 LET x8=x9: LET x$=""
9030 PRINT AT y9,x8; FLASH 1;"?"
;
9031 LET k$=INKEY$: IF LEN (k$)=
0 THEN GO TO 9030
9035 IF k$=CHR$ (13) THEN PRINT
k$: GO TO 9090
9040 IF k$<>CHR$ (8) THEN GO TO
9055

9050 IF x8<>x9 THEN LET x8=x8-1:
 LET x$=x$(1 TO LEN (x$)-1): PRI
NT AT y9,x8;CHR$ 32;: PRINT AT y
9,x8+1;CHR$ 32;: GO TO 9030
9051 GO TO 9080
9055 GO TO 9055+(k9*5)
9060 IF k$>CHR$ (44) AND k$<CHR$
 (58) AND k$<>CHR$ (47) THEN GO
TO 9085
9061 GO TO 9080
9065 IF k$>CHR$ (47) AND k$<CHR$
 (58) THEN GO TO 9085
9066 GO TO 9080
9070 IF k$>CHR$ (31) AND k$<CHR$
 (165) THEN GO TO 9085
9080 BEEP .5,.5: GO TO 9030
9084 PRINT AT y9,x8;k$;: LET x$=
x$+k$: IF x8>=t9 THEN GO TO 9090
9085 PRINT AT y9,x8;k$;: LET x$=
x$+k$: IF x8>=t9 THEN GO TO 9090
9086 LET x8=x8+1: PAUSE 50
9087 GO TO 9030
9090 IF k9<>3 THEN LET x=VAL x$
9095 RETURN
```



CODIGO MAQUINA

DESPIERTE CON SPECTRUM

También para despertarse puede resultar el Spectrum de gran utilidad, siempre que tenga un sueño ligero, o en caso contrario un sintetizador musical.

Como ya habrá adivinado, con este programa puede conseguir que su Spectrum actúe como una verdadera alarma. Para ello, sólo ha de introducir el tiempo que ha de transcurrir hasta que suene la alarma (en horas y minutos). Para activarla basta con llenar a la rutina en código máquina: RANDOMIZE USR 65120. Al transcurrir el tiempo fijado, el ordenador interrumpirá lo que estuviese haciendo para tocar una breve melodía, al mismo tiempo que el borde de la pantalla se volverá rojo.

Tenga en cuenta que la instrucción RANDOMIZE USR 65120 únicamente llama a la rutina para que se ejecute una vez. Cada vez que desee activar la alarma deber llamar a dicha rutina.

		Assembler	Comentario
10	Reloj	RST 56	
20		DI	
30		PUSH AF	
40		PUSH BC	
50		LD A,(65535)	
340	Alarma	LD A,2	Cambia color del BORDER.
350		CALL # 2298	
360		LD B,10	Número de BEEPs.
370	LI	PUSH BC	Guarda el valor de B.
380		LD A,1	Carga A con la duración de la nota.
390		CALL # 2028	Coloca A en el stack.
400		LD A,20	
410		CALL # 2028	Llama a la rutina de sonido.
420		CALL # 03F8	Segundo BEEP.
430		LD A,1	
440		CALL # 2028	
450		LD A,25	
460		CALL # 2028	
470		CALL # 03F8	
480		POP BC	Recupera B y completa la nota.
490		DJNZ LI	
500		LD A,62	
510		LD, I,A	
520		IM 1	
530	Fin	POP BC	
540		POP AF	
550		EI	
560		RET	

```
  10 CLS : CLEAR 65119: RESTORE
300
  15 PRINT TAB 13;"Relog";TAB 5;
 FLASH 1;"Cargando codigo maquin
a"
  20 LET sum=0: LET i=1
  30 READ a: IF a>255 THEN GO TO
 50
  40 LET sum=sum+a: POKE 65119+i
,a: LET i=i+1: GO TO 30
  50 IF sum<>a THEN GO TO 200
  60 LET sum=0: LET i=1
  70 READ a: IF a>255 THEN GO TO
 90
  80 LET sum=sum+a: POKE 65128+i
,a: LET i=i+1: GO TO 70
  90 IF sum<>a THEN GO TO 200
  95 CLS : PRINT AT 0,0;"Indique
 el tiempo que ha de       transcu
rrir para que suene la    alarma
(en horas y minutos)"
 100 INPUT "Hora?";h: POKE 65532
,h
 110 INPUT "Minutos?";m: POKE 65
533,m
 130 POKE 65534,0
 140 POKE 65535,0
 160 STOP
 200 PRINT "Error": LIST 9999: S
TOP
 300 DATA 62,9,237,71,237,94,201
,911,255,243,245,197,58,255,255,
254,0,40,6,61,50,255,255,24,97,6
2,49,50,255
 310 DATA 255,58,254,255,254,0,4
0,6,61,50,254,255,24,79,62,59,50
,254,255,58,253,255,254,0,40,6,6
1,50,253,255,24,61,62,59,50,253,
255,58,252,255
 320 DATA 254,0,40,6,61,50,252,2
55,24,43,62,2,205,155,34,6,10,19
7,62,1,205,40,45,62,20,205,40,45
,205,248,3,62,1,205,40,45,62,25,
205,40,45,205,248,3,193,16,226
 330 DATA 62,62,237,71,237,86,19
3,241,251,201,14404
```

PEEK POKE

RAPIDEZ UTILIDAD POTENCIA

VARIABLES DEL SISTEMA

Aprender a programar en BASIC es relativamente sencillo, lo cual no puede decirse del código máquina. Saber cómo trabaja el ordenador para hablarle en su propio lenguaje puede resultar un poco frustrante al principio. Por ello se suele empezar por averiguar el significado de esos PEEK y POKE que esporádicamente suelen aparecer en cualquier programa.

Unos se corren la voz... "¿sabes que el POKE 23692,255 evita el mensaje de SCROLL?" Otros, los más incautos, prefieren mirar el manual, creyendo que no hay nada mejor que un buen libro. Pero el manual de Sinclair, ya se sabe, adolece de buenas explicaciones, como es el caso del capítulo de las variables del sistema que trataremos exhaustivamente en este artículo.

De la dirección 23552 a la 23734 el Spectrum almacena una información muy valiosa que en muchos casos, y esto es lo importante, puede ser modificada.

Las variables del sistema son bytes en la memoria ROM del Spectrum que permiten al ordenador conocer los pasos que tiene que dar para ejecutar las órdenes que se le han dado. La información de cómo la memoria del Spectrum está repartida se guarda en estas variables, de forma que el ordenador pueda acceder a ellos y actualizarlos siempre y cuando lo necesite.

Se puede hacer uso de la información almacenada en estas direcciones, mediante su lectura o cambiándola para que el ordenador haga unas operaciones que de otra forma no haría o para que lo haga con más facilidad.

No todas las variables pueden resultar útiles. Y por supuesto no todas se pueden cambiar. Algunas causarán un bloqueo del ordenador o simplemente se ignorarán. Otras se pueden cambiar alegremente, pero sólo en circunstancias especiales y la mayoría se pueden variar, pero con unas condiciones y unas limitaciones muy estrictas.

Veamos cuál es la información contenida en estas variables y cómo sacar el mayor provecho de ellas.

23552-23559 (KSTATE)

Cuando el procesador se interrumpe una de las tareas que tiene que hacer es barrer el teclado y almacenar el resultado de dicha lectura en la variable KSTATE. Los bytes tienen usos diversos y no todos se pueden utilizar por el programador. El programa 1 ilustrará lo que ocurre en los 8 bytes de la variable KSTATE. Ejecútelo y pulse varias teclas para comprobar el efecto que causa el pulsar teclas individuales como SHIFT.

Los primeros 4 bytes de KSTATE tienen algo que ver con el *"two key rollover"*, que permite pulsar una segunda tecla antes de que se suelte la primera. PEEK 23556 devuelve el código del comando o letra situada en la parte superior de una tecla pulsada, de manera que si se pulsa SIMBOL SHIFT A, se tendría el valor del código de la letra "A" y no el de "a" ni el de STOP.

Esto puede ser útil cuando lo que interesa es teclear las opciones superiores de las teclas.

El efecto de pulsar una tecla es temporal y dura solamente mientras se mantenga pulsada. El valor almacenado en la dirección 23556 podría ser 255, si no se ha pulsado tecla alguna en el momento de la interrupción. Para la tecla ENTER dará el valor de 13. Para la tecla SPACE dará el valor 33, y pulsando ambas teclas SHIFT simultáneamente, se obtendrá el valor 14. El programa 2 lo demuestra.

23557 se utiliza para controlar el tiempo, previniendo así el contacto intermitente de una tecla.

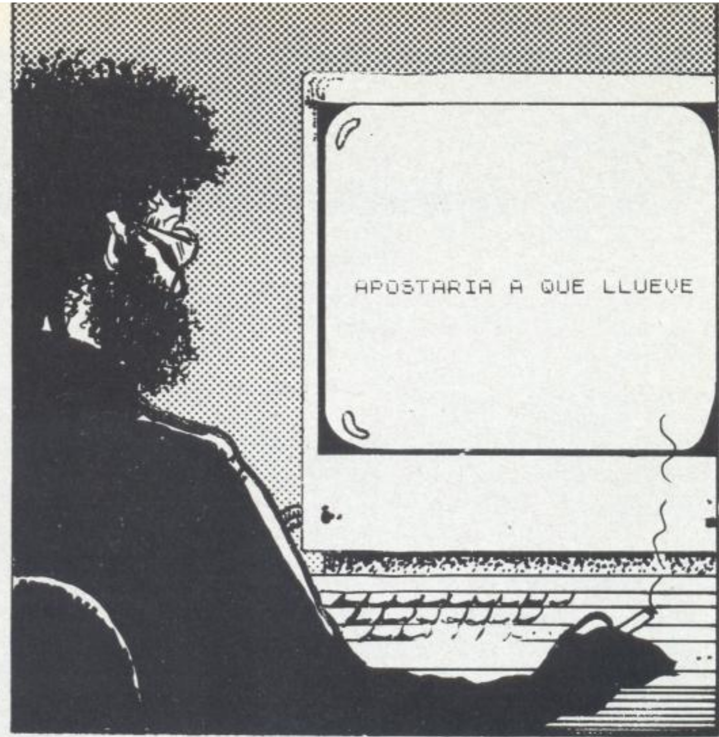
23558 es el que controla el tiempo de pausa antes de que la tecla comience a repetirse y el tiempo de repetición una vez que la tecla empiece a repetirse. El retardo utilizado está almacenado en las variables del sistema encargadas expresamente de ello, que son las que están situadas en los bytes 23561/2.

23559 contiene el código del último carácter pulsado. Este depende de si la tecla SHIFT ha sido pulsada o no. Los números producidos son los mismos que si se hubiera puesto

PRINT CODE INKEY$

excepto que el resultado sería la última tecla pulsada y no la que se pulsa en el momento. Pruebe el pro-






```
Programa num. 1
  10 FOR a=23552 TO 23559
  20 POKE 23692,0: REM Scroll
  30 LET b=PEEK a
  40 PRINT a;TAB 10;b;TAB 20;CHR
$ b AND b>31
  50 NEXT a
  60 GO TO 10
```

```
Programa num. 2

  10 LET a=PEEK 23556
  20 POKE 23692,0
  30 PRINT a,CHR$ a AND a>31
  40 GO TO 10
```

```
Programa num. 3

  10 LET a=PEEK 23559
  20 POKE 23692,0
  30 PRINT a,CHR$ a AND a>31
  40 GO TO 10
```

```
Programa num. 4

  10 PRINT "Pulse una tecla ahor
a"
  20 FOR a=1 TO 900
  30 NEXT a
  40 CLS
  50 LET a=PEEK 23560
  60 PRINT a: IF a>31 THEN PRINT
 CHR$ a
  70 GO TO 10
```

grama 3. Ejecútelo y pulse varias teclas utilizando conjuntamente la tecla SHIFT.

23560 (LASTK)

Cada vez que se hace un barrido del teclado y hay una tecla pulsada, el valor de esta variable se actualiza. Contiene el código de la última tecla pulsada.

Esta variable del sistema no hace nada que no se pueda conseguir con la instrucción INKEY$. En el programa 4 encontrará que si se pulsa una tecla cuando se lo indique el ordenador la respuesta aparece en la pantalla un tiempo después, aunque el programa se encuentre en la línea 50 cuando la pulsó. El código de la última tecla pulsada se almacena hasta que se pulse una nueva tecla. Es posible comprobar si se ha pulsado una nueva tecla examinando el quinto bit de la variable FLAG 23611. Su valor ha de ser uno si se acaba de pulsar una tecla.

Esta variable se puede utilizar para comprobar cualquier situación que requiera respuestas de tipo sí/no. Si se sabe que una situación así va a aparecer en un momento determinado de un programa, se podría adelantar la respuesta antes de que el programa llegue a dicha situación. Además, si dos teclas se pulsan simultáneamente, el programa respondería como si se hubiese pulsado una sola tecla.

Caracteres de control se pueden generar usando CAPS SHIFT junto con las teclas numéricas. La tecla ENTER dará el valor 13, pulsando las dos teclas SHIFT se obtendrá el valor 14. Para comprobar esto pruebe el programa número 5.

23561 (REPDEL)

Esta variable contiene la duración del tiempo que una tecla se puede mantener pulsada antes de que empiece a repetirse. Se puede alterar el valor con POKE, si se quiere que la

tecla comience a repetirse inmediatamente después de pulsarla. El cursor es difícil controlarlo si se hace POKE 23561,1. Al realizar POKE 23561,0 se elimina el efecto de repetición de la tecla y dando el valor 5 obtiene un retardo de cinco segundos.

23562 (REP PER)

Esta variable controla la duración del tiempo entre repeticiones una vez que la tecla pulsada comience a repetirse. Si se quiere apagar el efecto de repetición por cualquier motivo basta con hacer POKE 23562,0. Sin embargo, si se quiere aumentar el tiempo entre repeticiones hay que hacer POKE 23562,255. Si se quiere editar líneas de programas largas (como por ejemplo líneas PRINT con mucho contenido) haciendo POKE 23562,1 incrementará la velocidad del cursor. Pero hay que tener cuidado si se cambia con frecuencia, puesto que el cursor se puede descontrolar. Los valores normales con los que trabaja el cursor son 5/50 seg. ó 1/10 seg.

23563/4 (DEFADD)

Es la dirección del argumento de una función definida por el usuario en un programa. Así, si se tiene DEF FN A(B), el valor en 23563/4 será la dirección de la letra B en los paréntesis de esa línea, mientras que sólo se utiliza la función. La mejor forma de hacer PEEK en 23563/4 para ver esto es poner la instrucción PEEK como parte de la función FN a evaluar, ya que siempre valdrá cero, salvo cuando se evalúa la función. Lo que significa que la línea

```
10 PRINT PEEK 23563 + 256 ×
         PEEK 23564
```

siempre valdrá cero. Por otro lado

```
10 DEF FN A(B) = PEEK 23563 +
       256 × PEEK 23564
20 PRINT FN A(999)
```

contestará con la dirección de B en la línea 10. El valor 999 no es significativo, sólo es un dato que se da a B para prevenir un error. En el caso de que una función no tenga argumentos contestará con el valor de la dirección del paréntesis.

```
10 DEF FN A() = PEEK 23563 + 256
         × PEEK 23564
20 PRINT FN A()
```

23568 a 23605 (STRMS)

Los primeros 14 bytes en el BASIC del Spectrum contienen direcciones de los canales. Los canales —3 a +3 se guardan en dos bytes cada uno.

23606/7 (CHARS)

Esta variable tiene valores fijos: el cero para la 23606 y el 60 para la 23607. La variable apunta hacia la dirección del comienzo del juego de caracteres que utiliza para imprimir en pantalla e impresora. SCREEN$ también utiliza esta variable del sistema. La dirección normal a la que apunta es la 15360 cuyo valor es 256 menos que la dirección del comienzo del juego de caracteres de la ROM (256 menos porque al generador de caracteres se puede acceder con una instrucción del tipo PEEK 23606 + 256 × PEEK 23607 + CODE "A" × 8 y como el primer carácter es el espacio cuyo código es 32 se puede ver 8 × 32 = 256). El generador de caracteres ocupa 768 bytes, por lo que si se desea crear un nuevo juego de caracteres se debe dejar a un lado este número de bytes para no borrarlos.

Se ha mencionado que SCREEN$ usa esta variable. Realmente el problema reside en que SCREEN$ no reconoce los caracteres definidos por el usuario a no ser que éstos sean similares al que usa el Spectrum. SCREEN$ hace una búsqueda a través del juego de caracteres que está en una tabla hasta que encuentra uno similar.

Como el mapa de la pantalla del Spectrum se hace bit a bit en vez de

Programa num. 5

```
10 LET a=PEEK 23560
20 PRINT a,CHR$ a AND a>31
30 GO TO 10
```

Programa num. 6

```
10 FOR x=144 TO 164
20 PRINT AT 0,0;CHR$ x
30 POKE 23606,PEEK 23675
40 POKE 23607,PEEK 23676-1
50 PRINT AT 20,0;SCREEN$ (0,0)
60 PAUSE 40
70 POKE 23606,0
80 POKE 23607,60
90 NEXT x
```

Programa num. 7

```
8000 REM SCREEN$ PARA LOS UDG's
8010 LET a$=SCREEN$ (y,x)
8020 IF a$()="" THEN RETURN
8025 LET a=PEEK 23606: LET b=PEE
K 23607
8030 POKE 23606,PEEK 23675
8040 POKE 23607,PEEK 23676-1
8050 LET a$=SCREEN$ (y,x)
8060 IF a$<>"" THEN LET a$=CHR$
(CODE a$+112)
8070 POKE 23606,a
8080 POKE 23607,b
8090 RETURN


8070 POKE 23606,0
8080 POKE 23607,60
```

como lo sería un mapa de memoria de algunos ordenadores, una vez que se imprima algo en la pantalla permanece igual, aunque se cambie el carácter de la memoria. De manera que se puede alterar temporalmente el puntero del juego de caracteres hacia el juego de caracteres definidos por el usuario.

Aunque hay una variable del sistema que dice dónde comienza el juego de caracteres definidos por el usuario, esta dirección es 256 ó más, por lo que se debe restar 256. Esto significa que se debe restar uno del byte más alto. El programa 6 muestra cómo hacerlo.

De esta forma se logra que el ordenador crea que los nuevos caracteres sean los verdaderos. SCREEN$ seguirá produciendo caracteres con el código entre 32 y 127 aunque esto se puede arreglar trasteando un poco. Ya que SCREEN$ empieza con CHR$ 32 y el UDG empieza en 144, necesita sumar 112 para devolver el juego de caracteres situados en la zona de los UDG.

Esta es una forma de hacerlo, X es la coordenada x a través de la pantalla. Y e y son las coordenadas verticales que SCREEN$ tiene que examinar. Primero se chequea con SCREEN$ toda la pantalla en búsqueda de un carácter que no sea del juego normal de caracteres, y si lo hay vuelve a SCREEN$.

El carácter en X, Y se guarda en A$. La línea 8025 se necesita sólo si se utiliza un juego de caracteres que no sea el de la ROM. Si usa los caracteres de la ROM, entonces borre las líneas 8070 y 8080, y sustitúyalas por las dos líneas inferiores. (Ver programa 7).

Solamente hay 21 gráficos definidos por el usuario. Si SCREEN$ no encuentra su pareja continuará su búsqueda más allá de la zona dedicada a los UDG hasta que se salga del rango establecido por 32 y 127. Para evitar que esto ocurra, aunque los UDG se encuentran al final de la RAM, se puede añadir la siguiente línea:

```
6065 IF A$ > CHR$ 164 THEN LET
     A$=""
```

Hay que asegurarse que el contenido de 23606/7 siempre apunte hacia el juego de caracteres correcto, cuando se ejecuten instrucciones como PRINT, LIST, etc.

23608 (RASP)

Controla la duración del zumbido indicador de que se ha quedado sin memoria. Cuando se conecta el aparato esta variable tiene el valor 64. Esto se puede alterar pero tiene poco sentido. POKE 23608,0 hace que suene un *click* en vez del clásico zumbido, sin embargo POKE 23608,255 hace sonar un zumbido largo que bloquea el teclado evitando así que se introduzca más información.

23609 (PIP)

Controla la duración del *click* que suena al pulsar una tecla en el modo comando (K) o durante un INPUT. Comienza en cero, pero se puede alterar. Cualquier valor entre 30 y 130 hace que suene un *beep* más audible en vez del discreto *click*. Valores superiores a 130 hacen disminuir la velocidad de respuesta del teclado (debido a que no ejecuta hasta que no finaliza el *beep)*. El valor más corriente es POKE 23609,100.

23610 (ERR NE)

Controla el número que hay al producirse un error. Normalmente el 255, a no ser que se produzca un error, cuando esto ocurre el número que se almacena es una unidad inferior al que aparece en elmensaje de error. Por ejemplo, 4 Out of Memory guardaría un tres. El mensaje de error que aparece se halla situado a partir de la posición 5010 de la ROM. El final del mensaje se detecta al tener el séptimo bit del último carácter cambiado a uno. Después de los mensajes de error aparece © 1982 Sinclair Research Ltd. al igual que cuando conecta el ordenador o utiliza el comando NEW. Se puede hacer un POKE 23610 para generar un error que pare el programa, pero como el mensaje es fijo y está en la ROM, podríamos acabar con algo ilegible.

Si se quiere simular el error "Out of Memory" para acabar un programa basta con hacer POKE 23610,3. Esto puede formar parte del programa principal colocándolo al final del mismo con una de las condiciones de fin de programa.

23611 (FLAGS)

Esta variable contiene varias máscaras que controlan el BASIC y generalmente no se las debe alterar. De todo modos algunas máscaras pueden ser útiles.

Bit 0: Si es 1 indica que no se debe poner ningún espacio antes del próximo comando.
Bit 1: Si es 1, todo lo que aparezca con PRINT ha de volcarse hacia la impresora. Un cero imdica sacarlo por pantalla.
Bit 5: Cualquier tecla pulsada se indica por su código almacenado en 23560 (la variable del sistema LASTK) y el bit quinto de 23611 (FLAGS) se activa para indicar que una tecla se ha pulsado.
Bit 7: Máscara de sintaxis.

Son más útiles cuando se utilizan para programar en código máquina.

23613/4 (ERR SP)

Controla las direcciones donde se encuentran las pilas de RETURN. Intente hacer varios GOSUB sin poner RETURN alguno y compruebe este punto en la memoria. Ahora puede comprobar lo que ocurre cuando se queda sin memoria. También intente hacer un PEEK del contenido de estas tres direcciones (a cuya base apunta 23613/4 para ver con qué datos responde. (Ver programa 8).

23617 (MODE)

Especifica el modo de operación. Su valor puede ser 0, 1, 2 ó 4, según que el cursor sea L/C, E, G o K, respectivamente. Haciendo un POKE a esta variable afectará al modo del cursor. Puede aparecer como una

letra, número, símbolo o palabra clave parpadeante. Esto es más útil en una instrucción INPUT. El valor se inicializa cuando se necesite, por ejemplo, cuando se cambie el cursor desde el teclado.

Pruebe el programa 9 que realiza un POKE de todos los posibles valores en 23617. La mayoría son variantes de los cuatro modos de trabajo.

23618/9 NEWPPC Y 23620 NSPPC

23618/9 es una variable de dos bytes que contiene el siguiente número de línea a ejecutar. 23618 contiene el byte más bajo y 23619 el más alto, de manera que para leer el número de línea que contiene hay que hacer

PEEK 23618 + 256 × PEEK 23619

Para hacer un POKE de una línea determinada, por ejemplo X, se haría lo siguiente:

POKE 23618,X—256 × INT (X/256)
POKE 23619, INT (X/256)

Con las variables 23618/9 y 23620 se puede simular el comando GOTO hacia una línea de nuestro programa siempre que haga falta. El GOTO no se puede hacer a un lugar determinado de una línea con más de una instrucción.

Para saltar a una instrucción que se encuentra dentro de una línea muy larga del programa, por ejemplo, la cuarta instrucción de la línea X, primero habría que hacer todo lo descrito anteriormente y luego hacer POKE 23620,4 y saltará a dicha instrucción.

23624 (BORDCR)

Los bits de esta variable controla los atributos de la parte inferior de la pantalla y el color del BORDER. Haciendo varios POKES en esta variable, se puede conseguir que parpadee, brille o una pantalla inferior en multicolor, incluso se puede hacer que INK y PAPER sean del mismo color protegiendo de esa manera los programas. También se puede dar más brillo a los INPUTS para que resalte. (Ver tabla 1).

23627/8 (VARS)

Apuntan hacia las variables. Además de encontrar la zona de las variables, se puede ver la longitud que ocupa un programa con esta expresión. Esta excluye los bytes que ocupa la pantalla, las variables del sistema, las variables del programa y las pilas:

LET byte = PEEK 23627 + 256 × PEEK 23628 — PEEK 23635 — 256 × PEEK 23636

23629/30 (DEST)

Almacena la dirección de la variable cuando es asignada. Si es asignada antes, apuntará al principio de donde está situada la zona de variables. Si es definida por primera vez apuntará a la dirección del comienzo del nombre de la variable en el programa, por ejemplo: LET A=5, apuntará a la dirección de la letra A.

También puede ser útil para encontrar la dirección de memoria de una variable numérica. (Programa 10).

23631/2 (CHANS)

Almacena la dirección del comienzo del canal de información.

23633/4 (CHURCHL)

Almacena la dirección de la información de E/S utilizada hasta el momento. Normalmente apunta hacia un bloque de 5 bytes durante la operación de E/S. Utilice el programa 11 para examinar los contenidos.

23655/6 (PROG)

Contiene la dirección del comienzo del programa BASIC. Este apunta al primer byte del primer número de línea del programa. Puede ser útil a la hora de convertir programas a otros ordenadores de Sinclair Research con información almacenadas en las sentencias REM como primera línea de programa.

Si se quiere asegurar una línea de un programa bastará con hacer un POKE 0 en los dos primeros bytes del número de la línea, al comienzo del programa.

23637/8 (NXTLIN)

Guarda la dirección del comienzo de la siguiente línea de un programa. Esto es útil a la hora de acceder al

```
Programa num. 8

  10 LET a=PEEK 23613+256*PEEK 2
3614
  20 PRINT PEEK a;TAB 10;PEEK (a
+1);TAB 20;PEEK (a+2)
```

```
Programa num. 9

  10 FOR a=0 TO 255
  20 PRINT a
  30 POKE 23617,a
  40 INPUT a$
  50 NEXT a
```

```
Programa num. 10

  10 LET a=5
  20 LET a=a
  30 PRINT PEEK 23629+256*PEEK 2
3630
```

código máquina que se guarda en las sentencias REM en cualquier lugar del programa, por ejemplo aquellas subrutinas que se cargan con la instrucción MERGE desde una biblioteca de programas (Programa 11).

El único problema que puede generar es si se almacena dentro de una sentencia REM el color, INK, FLASH, BRIGHT pueden dar lugar a interpretaciones erróneas por parte del C/M. De todos modos si se utiliza la librería de subrutinas no habrá ningún problema puesto que no las utiliza.

23639/40 (DATADD)

Esta variable contiene la dirección de la última coma utilizada en una sentencia DATA. Si no se ha leído nada de la lista (por ejemplo, después de RUN) la dirección almacenada en 23639/40 es la del byte antes del área de programas, normalmente el CHR$ 128 al final del canal de información. Para confirmar esto ejecute el programa 12.

La dirección almacenada en esta variable de 2 bytes puede apuntar al carácter ENTER o a la coma dando a entender que ha finalizado una línea o sentencia que contiene DATA.

23641/2 (ELINE)

Esta variable contiene la dirección del área que se encuentra después de las variables. Desde aquí se puede hacer una idea de lo que ocupan la pantalla, las variables del sistema, el programa y las variables del programa, una vez que se ha ejecutado un programa fijando las variables. Teclee lo siguiente:

PRINT PEEK 23641 + 256 × PEEK 23642 — 16384

También puede obtener la información de la memoria ocupada por las variables una vez ejecutado el programa. Para ello teclee la siguiente instrucción:

PRINT PEEK 23641 + 256 × PEEK 23642 — PEEK 23627 — 256 × PEEK 23628

23653/4 (STKEND)

Contiene la dirección donde empieza la memoria libre. Leyendo estos bytes puede hacerse una idea de cuánta memoria queda libre restándola de la RAMTOP. Esto no incluirá la memoria utilizada para almacenar las pilas GOSUB, pero si la longitud de las sentencias PEEK. De manera que se puede orientar bastante.

PRINT PEEK 23730 + 256 × PEEK 23731 — PEEK 23653 — 256 × PEEK 23654

23658 (FLAGS 2)

Esta variable contiene algunas de las máscaras utilizadas por el ordenador para indicar ciertas condiciones.

La mejor forma de utilizarlo es comprobando el bit 3 que informa si CAPS LOCK está activado (1) o no (0).

¿Para qué sirve? Si en un programa se utiliza para la entrada de datos la tecla INKEY$ de manera que sólo admita las mayúsculas, con esta variable se obliga a la máquina a que todo lo que se teclee pueda ser entendido. El ordenador no razona como los humanos. Para usted es lo mismo decir "SI" (con mayúsculas) que "sí", pero no para la máquina. Es tentador el usar POKE 23658,8 para cambiar el estado del cursor y POKE 23658,0 para volverlo a su estado primitivo; por eso es conveniente comprobar el estado de las máscaras ya que éstas se pueden alterar sin que se dé cuenta. Normalmente, si el cursor está en el modo L, 23658 valdrá 0, por lo que no habrá ningún problema si se quiere utilizar esta variable.

23659 DF SZ

Esta variable contiene el número de líneas que hay en la parte inferior de la pantalla, normalmente utilizada para las entradas INPUT o los mensajes de error, etc. Normalmente valdrá dos salvo si tiene INPUTS muy extensos. Si hace POKE 23659,0 para limpiar la parte inferior de la pantalla y así utilizar el juego completo de las 24 líneas, lo único que conseguirá es que el sistema se venga abajo y se bloquee.

Sin embargo esto se puede hacer

Programa num. 11

```
   1 FOR x=0 TO 3: PRINT #x;PEEK
 23633+256*PEEK 23634: NEXT x: P
AUSE 0: STOP
23807
```

Programa num. 12

```
9000 LET a=USR (PEEK 23637+256*P
EEK 23638+5)
9010 REM Codigo Maquina
9020 RETURN
```

TABLA- 1

BIT 7	BIT6	BIT 5	BIT4	BIT 3	BIT 2	BIT 1	BIT Ø
FLASH	BRIGHT	BORDER Y PAPER			INK		

con algunas restricciones. Estas son que hay que restablecer el valor de esta variable antes de hacer cualquier uso de ella, de manera que hacer un BREAK en un programa puede ser catastrófico. También errores generados a lo largo del programa tendrán el mismo efecto, ya que los mensajes de error aparecen en dicha zona de la pantalla. El programa 13 ilustra lo dicho hasta ahora. Se pueden utilizar las dos líneas inferiores, la 23 y 24, con las instrucciones PRINT y PRINT TAB, pero no puede hacer PLOT, DRAW o PRINT AT, más allá de la línea 22. La pantalla se restablece haciendo POKE 23659,2 dentro del programa.

Para demostrar la clase de error que se puede generar, se puede "producir" un error añadiendo la siguiente línea:

45 PRINT error

Si se quiere utilizar las últimas líneas de la pantalla, lo mejor es usar la instrucción PRINT # 1; "texto" que funciona tan bien o mejor, que utilizando esta variable. Si cambia el valor de la variable por uno superior a dos, la parte superior de la pantalla se verá reducida más de lo normal. De manera que después de hacer POKE 23569,Y, la parte superior de la pantalla tendrá 24-Y filas y hará un scroll cuando la posición de PRINT llegue más allá de 24-Y,0.

El programa 14 demostrará cómo se puede mantener un scroll de la pantalla con la variable DF SZ y SCR CT. Aquí aparecen números aleatorios y hace un scroll de 14 líneas de la pantalla.

23670/1 (SEED)

Cuando se utiliza un número con la instrucción RANDOMIZE, ese número es almacenado en esta variable. Este es el número que determina el siguiente número aleatorio. Conociendo esto, se tiene la posibilidad de poder hacer "trampas", ya que se puede prefijar el número a salir en la próxima "tirada".

23672/3/4 (FRAMES)

Este es un contador de pantallas que se puede utilizar como cronómetro. Cuenta pantallas de TV e incrementa 50 veces por segundo. Aunque el tiempo que tarda en leer y evaluar estos tres bytes del reloj no permite que sea exacto. Tiene un límite máximo de cuatro días (unos tres días y 21 3/4 horas). El manual del Spectrum (capítulo 18) comenta que se necesita leer los valores de estos tres bytes dos veces consecutivas y coger el valor más alto de las dos lecturas para mayor seguridad. Hay que recalcar que el orden de los bytes es el inverso al que se está acostumbrado. El byte más significativo es el 23674, de manera que el valor del reloj hay que leerlo de la siguiente forma:

65636 × PEEK 23674 + 256 × PEEK 23673 + PEEK 23672

el cual contestará con el valor del tiempo en 1/50 de segundos.

Hay varias cosas que afectan la exactitud de este reloj: el comando BEEP para el reloj; la impresora y la gradora, etc., también afectan a dicha exactitud. Sin embargo, PAUSE no afecta en absoluto al correcto funcionamiento del reloj.

23675/6 (UDG)

La dirección de comienzo de los gráficos definidos por el usuario normalmente es 32000 en el Spectrum de 16K o 65368 en el de 48K. Este número es el mismo que el de USR "a", de manera que PRINT USR "a" se corresponde con:

PRINT PEEK 23675 + 256 × PEEK 23676

Los fanáticos del POKE se pueden divertir con esta variable. El manual aconseja cambiar esto para ahorrar espacio al tener menos gráficos definidos por el usuario. También se puede hacer todo lo contrario, es decir, establecer un juego de gráficos definidos aunque sólo se puedan utilizar un juego de 21 UDG a la vez. Recuerde que como se pueden hacer 21 UDG es necesario reservar 21 × 8 (168) bytes para cada juego separado de UDG y hacer un POKE del comienzo del juego de caracteres en la variable 23675/6.

Para divertirse teclee el siguiente comando:

POKE 23675,96 : POKE 23676,127 (Spectrum 16K)

POKE 23675, 96 : POKE 23676,255 (Spectrum 48K)

Entonces usando el juego de gráficos definidos por el usuario, intente teclear cualquier mensaje... ¡ya verá lo que ocurre!

Sólo le daremos una pista, una vez que establezca el juego de UDG guárdelo en cinta. La mayoría de los usuarios hacen algo parecido a SAVE "carac" CODE 32600,168. Esto está bien, pero obliga a especificar la dirección del comienzo del juego de UDG. También se puede hacer SAVE "carac" CODE (PEEK 23675 + 256 × PEEK 23676),168 y ya puede guardar

```
Programa num. 13

  10 DATA "1","2","3","4","5"
  20 LET a=PEEK 23639+256*PEEK 2
3640
  30 PRINT a;TAB 9;PEEK a;TAB 18
;CHR$ PEEK a AND PEEK a>31
  40 READ b$
  50 GO TO 20
```

```
Programa num. 14

  10 POKE 23659,0
  20 FOR a=0 TO 23
  30 PRINT a
  40 NEXT a
  50 PAUSE 0
  60 POKE 23659,2
```

el actual juego de gráficos que haya definido en cinta sin preocuparse de la dirección de donde empieza. Esto por ejemplo, permite cargar un juego de caracteres que se haya definido en un Spectrum de 16K a otro de 48K sin tener que conocer las direcciones.

Para cargar un juego de UDG en otra máquina con distinta memoria basta con hacer

LOAD "carac" CODE (PEEK 23675 + 256 × PEEK 23676),168

Esto automáticamente relocalizará los datos en la dirección correcta, para que la máquina haga uso de ellos cuando los necesite. Esto sería lo mismo que LOAD "carac" CODE USR "a", pero lo único que ahorra es el teclear tanto.

más alto de la dirección del *buffer* de LPRINT. Este se puede variar a gusto del programador, pero si se utiliza la impresora se volverá a reinicializar a 91. 23680 y 23681 juntos contienen la dirección de la posición de LPRINT dentro del buffer de la impresora. No afectará de ninguna manera si se hace algún POKE a la impresora, pero cualquier cosa que se le almacene se borrará si se utiliza la impresora.

23677/8 (COORDS)

23677 es la variable que contiene la coordenada x del último punto PLOT. Después de hacer CLS comienza en cero y 23678 es la variable que contiene la coordenada Y del

23688/9 (SPOSN)

23688 contiene información acerca de hasta donde ha llegado la sentencia PRINT en la pantalla. Comienza en 33 a la izquierda de la pantalla y va decreciendo en una unidad cada vez que la posición PRINT se dirige a la derecha. Después de utilizar la instrucción RPINT AT X,Y (si X e Y son coordenadas válidas) 23688 almacenará el valor 33—X. Esto puede servir para evitar que las palabras se corten al final de una línea de forma poco ortodoxa. Imagínese que el número contenido en 23688 va descontando hacia cero según se va llenando la línea, podemos ver que comparando esto con la longitud de la palabra a

TABLA- 2

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT Ø
PAPER 9 perman.	PAPER 9 temp	INK 9 perman.	INK 9 temp.	INVERSE 1 perman.	INVERSE 1 temp.	OVER 1 perman.	OVER 1 temp.

23679 (P POSN)

Contiene la información acerca de la longitud que tiene que recorrer la instrucción LPRINT en el buffer de impresora. No se puede variar el contenido con POKE.

23680 (PRCC)

Contiene el byte más bajo de la dirección a donde irá situado el siguiente carácter en el *buffer* de la impresora, si es cero irá situado en la columna izquierda, 15 si va en la columna 15, etc. Como esta es la dirección de la línea superior de cada carácter, se puede variar (POKE) para cambiar la posición LPRINT en el *buffer* siempre y cuando se cambie a su vez el valor situado en la variable P POSN (23679) para igualarlo.

23681. Variable del sistema sin asignar

Esta variable aunque no se utiliza, siempre contiene 91. Esto es el byte

último punto PLOT. Contiene el valor actual, de manera que si el último punto PLOT tenía como coordenadas 3,3 ambos bytes almacenarán un tres.

Estos valores se les puede variar con POKE, ya que ello implica el dibujar dicha coordenada en pantalla. Esta es una manera de mover el cursor PLOT alrededor de la pantalla. Esto se puede hacer mediante la instrucción

PLOT OVER 1; X, Y: PLOT OVER 1; X,Y

pero sería un poco engorroso.

23684/5 (DF CC)

Almacena la dirección del fichero de salida de la posición PRINT. Se le puede variar para mandar los PRINT hacia otra dirección, aunque esto requiere una cierta preparación de cómo está organizada la pantalla.

escribir se puede saber si la palabra cabe en esa línea o hará falta saltar a la siguiente para evitar que la palabra sea cortada en el momento de que 23688 valga cero. Supongamos que la palabra a escribir sea W$, la instrucción sería la siguiente:

IF PEEK 23688 < LEN W$ +1 THEN PRINT

Esto sólo funciona con palabras cuya longitud sea inferior a 32 caracteres.

23689 contiene la información relativa al número de líneas que le queda por rellenar en la pantalla. El valor inicial es 24 que es el de la línea superior de la pantalla y decrece en una unidad según se va bajando por la pantalla. Si en vez de querer hacer un scroll cada vez que se llega a la parte inferior de la pantalla, lo que quiere es limpiarla, bastará con hacer la siguiente instrucción:

IF PEEK 23689 = 3 THEN CLS

23692 (SCR CT)

Contiene el número de *scrolls* que tiene que hacer más uno antes de preguntar si hace otro *scroll*. En los juegos gráficos esto puede generar problemas, ya que no interesa esperar tiempo de respuesta. De manera que si en la variable 23692 se almacena cualquier número que no sea uno, no hay tiempo de espera. Por lo que con POKE 23692,255 el ordenador escribirá 255 líneas en la pantalla antes de preguntar si hace un scroll.

23693 (ATTR 0)

Contiene los atributos permanentes o los atributos globales. Colores particulares en las sentencias PRINT se manejarán en otra parte. La mayoría de las rutinas de la ROM utilizan los valores de la variable del sistema con un valor temporal, ya que éstos contienen unos atributos permanentes a no ser que se varíen. CLS limpia la pantalla y la pone como establezca la variable ATTR P. La función de los bits es la siguiente:
Bit 7: 1 cuando FLASH es 1.
Bit 7: 0 cuando FLASH es 0.
Bit 6: 1 cuando BRIGHT es 1.
Bit 6: 0 cuando BRIGHT es 0.
Bits 5, 4 y 3 contienen el color del papel en binario, por ejemplo, para PAPER 7, los bits 5, 4 y 3 valdrán todos 1.
Bits 2, 1 y 0 contienen el valor de INK en binario, al igual que PAPER.

Atributos como el 8 y el 9 no se manejan con esta variable. Si el valor del atributo fuera 8 ó 9, entonces los valores establecidos por esta variable no serían válidos.

23694 (MASK P)

Esta variable ayuda al Spectrum a determinar el atributo de cualquier parámetro cuyo valor sea ocho. De manera que si especifica BRIGHT 8 globalmente, el bit 6 de 23694 valdría uno para recordarle a la máquina en el futuro que BRIGHT 8 ha sido especificado. Para determinar el color/*flash*/*bright* cuando se imprimen mensajes el ordenador comprueba lo que hay allí e imprime la palabra con esa característica. O si se quiere puede hacer una sobreimpresión del carácter en la pantalla dejando los atributos tal y como están. (Ver tabla 2).
Bit 7: 1 cuando FLASH 8 está activado.
Bit 6: 1 cuando BRIGHT 8 está activado.
Bits 5, 4 y 3 valen todos 1 cuando PAPER es 8.
Bits 2, 1 y 0 valen todos 1 cuando INK es 8.

23695 (ATTR T)

Esta variable contiene el valor actual del color establecido por una sentencia PRINT. Esto lo puede ver por sí solo con las siguientes instrucciones:

PRINT PEEK 23695
PRINT INK 7: PAPER 0: PEEK 23695

Esto es, se incluye el comando PEEK en la instrucción PRINT bajo el efecto de un color de tinta. Normalmente, a no ser que se especifiquen valores particulares, esta variable almacenará el valor global del color. Los atributos utilizados para imprimir se toman de esta variable y las cosas se equilibran de manera que la única diferencia entre las variables ATTR T y ATTR P es que en aquella los valores de los atributos decrecen.

23696 (MASK T)

Es similar a la variable MASK P (23694) excepto que los parámetros aquí son temporales. Normalmente el mismo que el parámetro permanente de ocho, éste cambia mientras los colores locales, etc. efectúan su cometido. Se verá con más facilidad con el siguiente comando directo:

PRINT PEEK 23696, INK 8: PEEK 23696, INK 0: FLASH 8: PEEK 23696

Los bits individuales tienen las funciones que se ilustran en la tabla 5.

23697 (P FLAGS)

Esta variable contiene las máscaras necesarias utilizadas durante el proceso de imprimir. Especificando PAPER 9 los bits 6 y 7 son puestos a 1. Con INK 9 son puestos a 1 los bits 4 y 5. Con INVERSE 1 los bits 2 y 3 son puestos a 1. Y después de que se especifique OVER 1 se ponen a 1 los bits 0 y 1. Los efectos son permanentes si se ponen a 1 los bits impares y temporales si se ponen a 1 los bits pares. (Ver tabla 6).

23681/23728/9

Estos tres bytes no se suelen utilizar por el Spectrum. Pueden ser utilizadas en los programas par almacenar información que interese. Estos bytes son muy útiles a la hora de programar en lenguaje máquina pudiendo acceder a ellos directamente en vez de buscar la variable en el área de las variables. 23728/9 se pretendían utilizar para interrupciones sin máscara, pero éstas no existen en el Spectrum.

```
Programa num. 15

  10 POKE 23692,0: POKE 23659,10
  20 PRINT RND
  30 GO TO 10
```

```
Programa num. 16

  10 INK 8
  20 POKE 23694,BIN 00000011
  30 PRINT AT 0,0; INK 5;"555555
5"
  40 PRINT AT 0,0;"1111"
```

TABLA- 3

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT Ø
FLASH 8 Temp.	BRIGHT 8 Temp.	PAPER 8 Temp.			INK 8 Temp.		

23730/1 (RAMTOP)

Esta variable de dos bytês apunta hacia el último byte de la RAM del área del BASIC. Esta no indica el final de la memoria utilizada por el BASIC en el sentido de que los UDG normalmente se almacenan por encima de esta dirección. Si se sube la RAMTOP por encima de la zona dedicada a los UDG se pueden perder éstos, pero a su vez se ganará memoria, que puede ser útil sobre todo para aquellos usuarios con el Spectrum de 16K.

Un punto importante es que el comando NEW opera hasta la zona de memoria indicada por la variable 27320/1, de manera que se puede almacenar DATAS en estas direcciones para que sean utilizadas por distintos programas.

23732/3 (RAMPT)

Finalmente, esta variable contiene la dirección donde acaba la RAM del Spectrum. Si se adquiere un Spectrum cuya capacidad de memoria nos es desconocida, no hay por qué mirar sus interioridades, bastará con teclear el siguiente comando:

PRINT PEEK 23732 + 256 × PEEK 23733 — 16384

Los 16384 bytes restados son para la ROM, ya que la RAM comienza a partir de esta dirección y va hasta la dirección indicada por la variable almacenada en P RAMT.

La Proteción del Software

E terreno de la informática es aún muy reciente, y como ocurre siempre en estos casos existen muchas lagunas legales por cubrir. Una de estas ausencias más importantes y reiteradamente destacadas por los profesionales de la informática es la ausencia de leyes de protección de *software*. Las medidas de protección de la propiedad intelectual actuales son tan obsoletas como inoperantes. Ante esta situación los "ingenieros del *software*" investigan nuevos procedimientos y nuevas combinaciones *hardware-software* para evitar, o al menor complicar, la piratería de su trabajo.

En ocasiones lleva incluso a nuevos procedimientos comerciales. En el lanzamiento del nuevo QL, Sinclair incluye en su precio cuatro potentes programas desarrollados por Psion, consciente de que de otra forma las copias serían inevitables (y esto en un país como Inglaterra donde el control de la propiedad intelectual está bastante regulado).

Ningún procedimiento de protección es totalmente infalible, lo que lleva a una espiral de sofisticación constante de estos procedimientos. Esperamos con este artículo poner algunos de estos medios en sus manos para que también usted pueda proteger sus propios programas. Algunas de estas técnicas son utilizadas comúnmente por los profesionales del *software*.

El primer problema empieza con la carga de programas. Si el programa no empieza a ejecutarse automáticamente será imposible evitar que el usuario introduzca el comando SAVE. Para ello el Spectrum dispone de la instrucción SAVE "nombre" LINE XXXX con la que el programa cargado se ejecuta inmediatamente a partir de la línea especificada. Este es sólo un primer paso, bastante inútil, si es el único, pues el Spectrum también tiene la instrucción MERGE con la que poder coger un programa a fin de unirlo a otro, que nunca será ejecutado, aunque hubiese sido grabado utilizando la instrucción LINE. Sinclair ha mejorado este aspecto con el lanzamiento de los *Microdrive*, con los que no es posible utilizar MERGE con aquellos programas en que se utilizó la opción LINE. Pero con la cinta sí es posible.

A primera vista puede parecer que SAVE "nombre" LINE es la única forma de lograr que el programa se ejecute inmediatamente después de haberse cargado. Examinando los programas comerciales puede comprobarse que programas grabados en código máquina también se ejecutan de forma automática. El Spectrum permite grabar cualquier área de memoria con la instrucción SAVE "nombre" CODE, COMIENZO, NUMERO BYTES. Grabando toda la información contenida en la RAM se graba el programa en BASIC. Cuando se efectúe la carga de los bytes grabados, el ordenador estará en idéntica situación a cuando se efectuó la grabación. El Spectrum almacena el BASIC en un área de memoria que dispone al efecto, utilizando 180 bytes para descripción del programa (variables utilizadas, pri-

mera instrucción a ejecutar, etc.). El siguiente programa quedará grabado de esta forma. Ejecútelo, desconecte su ordenador o introduzca la instrucción RANDOMIZE USR 0 y cargue el programa (LOAD "DEMO"...)

```
100 PRINT "Antes del SAVE"
110 SAVE "DEMO" CODE 23552,
1500
120 PRINT "Después de SAVE"
130 GOTO 120
```

Como verá, acabamos de encontrar otra forma de que un programa en BASIC se auto-ejecute sin necesidad de recurrir al comando LINE, evitando el uso de MERGE. De hecho esta es una técnica ampliamente utilizada en el *software* comercial.

Cambiando el valor 23552 por 16384 en la línea 110 se logra grabar el programa en BASIC y la imagen de la pantalla en el momento de efectuar la grabación, obteniéndose posteriormente el gráfico con gran rapidez.

Desgraciadamente aún es posible entrar en el listado, pero es más difícil, sobre todo si es largo. El comando LOAD "nombre" CODE permite especificar una nueva localización de memoria a la que entrar posteriormente. Lógicamente, es más difícil en un Spectrum de 16 K.

Además, otro de los inconvenientes de utilizar este procedimiento es la imposibilidad de verificar la grabación, ya que el comando VERIFY siempre nos dará error, aun cuando la grabación haya sido correcta. Esto es debido a la grabación de datos que están continuamente cambiando, como por ejemplo el reloj. Algunos programadores utilizan esto para descubrir las copias. Conociendo el valor del reloj en el momento de la grabación se puede después comparar cuando en el programa se ejecute. Una pequeña diferencia evidencia la existencia de una copia y el programa se autodestruye. Pero tampoco esto es infalible: una vez que se logra entrar en el listado, sólo hay que encontrar la instrucción que busca el valor del reloj, es decir, buscar el valor 32673 y producir un salto sobre esta instrucción. El capítulo 18 del manual ofrece un buen ejemplo sobre el uso del reloj.

Pero no nos desanimemos por ello. Otra posibilidad estriba en eliminar la comunicación con el usuario o, lo que es lo mismo, en desactivar el teclado. El Spectrum comprueba si es presionada alguna tecla, entre otras cosas, 50 veces por segundo. Si eliminamos este control, eliminaremos la posibilidad de interrumpir el programa al mismo tiempo que se logrará que éste vaya más rápido. Veamos cómo:

```
100 POKE 23296,243
110 POKE 23297,201
120 RANDOMIZE USR 23296
```

Pero recordemos que el reloj es para el Spectrum lo que el corazón para las personas: no puede vivir sin él. El ordenador no responderá cuando se presione alguna tecla, con la excepción del comando INKEY$. Introduzca dos líneas más al programa antes de ejecutarlo:

```
130 PRINT INKEY$
140 GOTO 130
```

Este pequeño truco puede ser interesante, pero ha de utilizarse con precaución. Existen además determinadas funciones que son controladas por el reloj. Por ejemplo, BEEP, SAVE o COPY, que lógicamente no podrían llevarse a cabo si se le desactiva. La principal ventaja que ofrece es de rapidez, al evitarse que el ordenador compruebe 50 veces por segundo si alguna tecla ha sido presionada.

Si seguimos investigando la forma en que podemos evitar que alguien acceda al listado, pronto pensaremos en la posibilidad de evitar el uso de la tecla BREAK o, al menos, hacer que funcione de otro modo. Tanto el mensaje de BREAK como STOP son tratados por el Spectrum como un error. Veamos cómo trata el Spectrum a los errores.

La variable del sistema ERR-SP (dirección 23613-4) se encarga de los errores. Realmente en esta dirección se incluye la dirección de otra localización a la que el ordenador saltará para el oportuno mensaje de error. ¿Ha pensado ya que ocurriría si cambiásemos este valor? El siguiente programa obliga a bifurcar a la dirección 0 cuando ocurra un error, con lo que se autodestruirá el programa

```
100 POKE 23613,2
110 POKE 23614,91
120 POKE 23298,0
130 POKE 23299,0
140 PRINT "?";
150 GOTO 140
```

El programa direcciona ERR-SP hacia las direcciones 23298 y 23299, normalmente utilizadas por la impresora donde se ha colocado la nueva dirección (O). También puede elegir otras direcciones distintas del *buffer* de la impresora y otras bifurcaciones distintas de cero. Si siempre desea el salto al inicio de la ROM (dirección 0) puede referenciarlo directamente a una posición de memoria que bifurque a 0:

```
100 POKE 23613,4
110 POKE 23614,61
```



Pero tampoco este método resulta del todo infalible. La variable del sistema ERR-SP puede ser alterada según el desarrollo del programa. Comandos como RUN, GOSUB, RETURN o CLEAR cambian su valor, por lo que ha de elegir entre no utilizarlos o buscar otro método. No obstante, este método puede resultar bastante eficaz en pequeñas rutinas.

El Spectrum tiene un *display* muy original, dividiendo la pantalla en dos zonas. La parte superior donde aparecen los textos y gráficos y la parte inferior para la introducción de datos y mensajes de error. Es fácil deducir que si se elimina esta segunda parte el programa podrá seguir trabajando sin que se altere y sin que, por tanto, se pueda hacer uso del comando SAVE que necesariamente pasaría por la zona inferior de la pantalla.

En la dirección 23659 se encuentra la variable DF-SZ. Según informa el manual se utiliza para almacenar el número de líneas de dicha zona, normalmente dos o más. Colocar un cero (POKE 23659,0) en un programa es lo mismo que eliminar esta zona. Una vez más impedimos la interrupción del programa al presionar BREAK, pues el ordenador quedará confundido (léase bloqueado) al no tener líneas libres en pantalla para mostrar el mensaje correspondiente. El único problema es que afecta a los comandos CLS y BORDER, pues ambos inciden en las dos zonas de impresión de la pantalla. Al intentar usar uno de estos dos comandos el ordenador quedará bloqueado. Puede pensarse en desbloquear antes de utilizar estas instrucciones y volver a bloquear después, pero resultaría excesivamente tedioso. Afortunadamente no es difícil desarrollar una rutina que simule los efectos de BORDER y CLS:

```
8000 OVER O: INVERSE O
8010 FOR Y=O TO 21
8020 PRINT AT Y, O.,
8030 NEXT Y
8040 RETURN
8100 LET C=BORD*8
8110 FOR Y=23232 TO 23295
8120 POKE Y, C
8130 NEXT Y
8140 OUT 254,BORD
8150 IF C<=32 THEN LETC=C +7
8160 POKE 23624,C
8170 RETURN
```



En vez de usar CLS y BORDER puede utilizar GOSUB 8000 y GOSUB 8100, respectivamente.

La primera rutina es bastante sencilla e ingeniosa, eliminando la información de cada posición, pero dejando el color existente. Las dos comas del final de la línea 8020 permiten "barrer" los 32 caracteres por línea (16 caracteres cada línea).

La segunda subrutina no es tan sencilla. Primero realiza una conversión del color al código correspondiente, de acuerdo a la tabla de atributos descrita en el manual. Este valor se almacena. El siguiente paso es cambiar el color del BORDER. Para ello la línea 8140 envía un nuevo valor de la ULA. Después, con ayuda de la variable del sistema BORDER (posición 23624) y de la línea 8160 colocamos el nuevo color.

Este método es de los más utilizados ya que no presenta demasiados problemas, como veíamos con los anteriores.

Sea cual sea el método elegido, todo programa puede ser copiado. En el terreno de la duplicación del *software* existen dos procedimientos: copia del *cassette* a *cassette* y copia bit a bit.

CASSETTE A CASSETTE

Es un proceso simple en el que el único requisito es disponer de dos

LO QUE OPINAN LAS CASAS COMERCIALES

La realización de un buen programa comercial no termina con la verificación de su correcto funcionamiento. Posteriormente habrá de dotársele de las últimas innovaciones en materia de software de protección. Siempre hay alguien que logra vencer los obstáculos más difíciles, entrándose en una espiral que lleva a una enorme sofisticación en los dispositivos de protección empleados. El uso del cassette ha permitido una gran difusión de la informática al ser un medio económico para almacenamiento de programas, pero se vuelve ahora contra sus creadores, quienes no ven soluciones claras al problema de la piratería del software distribuido en cassettes. Así nos lo contaba José Luis Domínguez, Director Gerente de Indescomp, una de las pocas casas de software que realizan programas para el Spectrum reconocidos internacionalmente. "Es un problema de muy difícil solución. Los ingleses —que son los grandes maestros— no han conseguido resolver el problema, hasta tal punto que ya han quebrado varias empresas."

"Hemos estado seis meses intentando que la gente entendiera que es el software."

El mercado español todavía no está lo suficientemente desarrollado. Es por ello que la típica piratería industrial todavía no haya entrado o al menos José Luis no está muy convencido de que llegue por el momento. "En Inglaterra sí merece la pena fusilar los best-seller porque hay un mercado muy amplio. De uno de nuestros programas —El Fred— aparecieron 30.000 copias en un almacén, y esto es sólo lo que se localiza esporádicamente. También sabemos que hay algunos comerciantes que hacen copias de juegos para regalarlos con la venta de los Spectrum, pero eso no nos preocupa. Lo realmente preocupante es la piratería del aficionado. Un chaval se compra el Bugaboo, se lo lleva al colegio y se hacen diez copias. Está claro que esos diez chavales eran diez potenciales compradores de esa cinta y ahora no lo son. Es un problema que en el futuro se va a plantear muy seriamente".

Desde el punto de vista legal poco o nada se puede hacer. En Inglaterra ha hay varias asociaciones para luchar y combatir la piratería. En España todavía hay que vérselas con el funcionario de turno para tratar de explicarle lo de la propiedad intelectual, como

cassettes, para grabar directamente de uno a otro como si del éxito del último *single* se tratase. Los expertos en la materia aseguran que cuanto peores son los *cassettes*, mejor. Pero la verdad es que la calidad deja mucho que desear.

BIT A BIT

Es un secreto a voces la existencia de un programa llamado "la llave" por aquello de que abre todas las cintas. Se trata de un programa que lee datos de una cinta sin interpretarlos. Los primeros bits suelen determinar el tipo de programa, su nombre, etc. Después siguen todos los que configuran el programa. De esta forma se logra una copia tan perfecta como el original.

CONCLUSION

Esperamos que este artículo le haya sido útil para proteger sus programas adecuadamente, aunquenos tememos que será igual de útil a aquellos cuya afición favorita es desproteger programas. En cualquier caso, la polémica sobre la protección de propiedad intelectual está por empezar y sólo se nos ocurre un medio infalible para evitar las copias: el abaratamiento del *software*.



nos contaba anecdóticamente: "Nos hemos vuelto locos durante seis semanas para registrar la propiedad intelectual, tratando de hacerles entender qué era eso del software". La ley llegará con retraso, como siempre. Mientras tanto los ingenieros del software ponen a prueba sus neuronas para la realización de nuevas rutinas. "Utilizamos subrutinas en código máquina que hacen que sea muy difícil abrirlo. Pero siempre hay alguien que inventa una rutina que destruye la rutina anterior. En una ocasión hicimos un cassette especial, cortando la cinta en una medida exacta. Al llegar al final quedaba una instrucción que saltaba, pero tampoco era ésta la solución. Realmente no hay soluciones válidas mientras se trabajecon cassette". Con el microdrive se podrán conseguir mejores sistemas de protección, pero su precio no parece apuntar a un amplio sector del mercado. La cosa parece clara, hay que utilizar un sistema distinto al cassette, pero que no resulte tan caro como el microdrive. José Luis nos revelaba su secreto: "Tenemos ya

"O se busca una solución concreta o la industria del programa en cassette va a correr un grave peligro."

un prototipo de cartucho que estamos desarrollando con una empresa americana y que esperamos presentar en noviembre. Será un poco más caro que la cinta, pero podrá competir con ella y con las ventajas de un cartucho (carga inmediata, evitar piratería, etc.). Se trata de un cartucho muy especial, pues se podrán cargar en ellos nuevos programas. Cuando uno se cansa de un juego se va a la tienda y se cambia por otro. Una máquina que estamos desarrollando permitirá hacer una copia del nuevo programa en el cartucho viejo."

"Está claro que nos copian muchísimo", reconocía José Luis Domínguez. Pero está también claro que el desarrollo del parque informático en España traerá nuevos desarrollos de protección hardware-software. Quizás este prototipo de cartucho sea sólo el comienzo. Por el momento la piratería de software en España no deja de ser un problema menor para las casas de software, quienes se muestran particularmente preocupadas por el futuro, como sentenciaba finalmente José Luis: "O se busca una solución concreta o las industrias del programa en cassette van a correr un grave peligro."

Sintonice bien su Spectrum

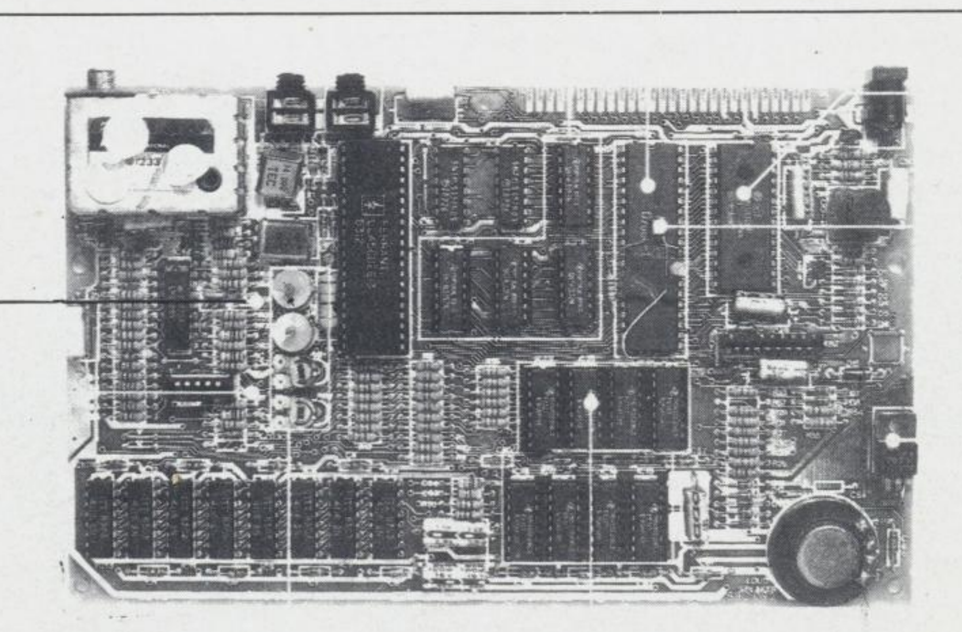
Con los ordenadores, al igual que ocurre con algunos automóviles, suele hacerse referencia siempre al modelo y a la marca. Así se oye hablar del ZX81 o del Spectrum, ambos productos de Sinclair Research (y a veces se observa una cierta confusión entre ambos modelos, tan diferentes, cuando alguien dice: "tengo un Sinclair"). No siempre dos ordenadores con el mismo nombre y apellido son iguales, pero es obvio que el *software* que corre en ellos lo hace sin problemas. Esta circunstancia explica que para muchos usuarios haya pasado desapercibido el hecho de que existen no una ni dos sino tres versiones sucesivas del Spectrum. En España sólo se han comercializado las dos últimas.

Si es usted una persona poco amiga de crearse problemas, no se preocupe: tanto la versión 2 como la 3 realizan las mismas funciones. Pero si es usted curioso y le gusta meterse con la circuitería, puede que le haga ilusión echar una mirada a las entrañas de su Spectrum. Compruebe, en base a las fotografías que le ofrecemos en este artículo, cuál es la versión que usted posee.

La existencia de varias versiones no es cuestión de capricho, mucho menos en una máquina tan pensada como es el Spectrum. Se trata, como siempre, de ir incorporando a la idea original los nuevos avances tecnológicos. En este artículo sólo prestaremos atención a un aspecto: la posibilidad de sintonizar el Spectrum lo mejor posible con el televisor. Claro que, a esta altura, conviene aclarar que lo que aquí se diga valdrá para la segunda versión, ya que la tercera permite realizar la sintonía de forma automática.

Una de las peculiaridades de los dos primeros modelos es la flexibilidad del *display*. Flexibilidad que se torna en problema si lo que se desea es conectarlo a un monitor. Pero esta flexibilidad le permitirá poder realizar unos mejores ajustes. Para ello ha de empezar por quitar la carcasa para poder trabajar con la placa, siendo ésta su primera decisión importante, ya que con ello se invalida toda posible garantía. Si ya ha decidido dar este primer paso, retire los tornillos situados en la parte inferior del Spectrum; suelte los dos cables planos a los que está conectado el aparato con sumo cuidado, ya que no hay longitud suficiente como para desen-

VERSION 2



El cable "VID" (video). El interface del monitor del que Sinclair estaba muy orgulloso solamente lo hallará en las versiones modificadas del Mark 1 y 2. Cable plano en color del Teletext: Y, V y U.

volverse con tranquilidad: localice los condensadores y resistencias variables situados en los alrededores del modulador de televisión y ya estará en disposición de hacer el ajuste, siendo distinto según tenga conectado el ordenador a un televisor o a un monitor.

Con Televisor

Hay varios componentes que merecen la pena conocerse antes de proceder a la sintonización del aparato. Además en la zona del modulador encontrará el VID, que se analizará en su momento ya que tiene que ver con el ajuste del monitor. Los condensadores variables son TC1 y TC2 y las resistencias variables son VR1 y VR2. Trabajando de arriba a abajo esto es lo que hace cada uno de ellos:

TC1 controla el brillo.

TC2 sintoniza los colores de E/S.

VR1 ajusta los colores azul/amarillo.

VR2 ajusta los colores rojo/verde.

Antes de continuar, asegúrese de que trabaja con un destornillador que no sea metálico y por supuesto tenga cuidado a la hora de apretar o soltar, puesto que hay que hacerlo con el aparato encendido para poder ver los resultados de la sintonización en la pantalla. Para empezar, ajuste el tornillo de TC2 con cuidado en ambas direcciones hasta que encuentre el tono más fuerte del color, aunque éste sea débil. Esto se hace ajustando los colores con el tornillo situado en VR1. Una vez localizado el mejor color hacer lo mismo con el tornillo VR2. Lo esencial reside en combinar estos tres tornillos para tener la imagen lo más clara posible.

TC1 es el condensador que controla el brillo en la pantalla. Una vez que el color está a su gusto, este es el tornillo a fijar. De todos modos el brillo puede interferir con los demás controles por lo que habrá de ir ajustándolos poco a poco. Una vez completado los delicados ajustes deberá tener una imagen clara.

Con Monitor

Hasta aquí todo está muy bien para aquellos que quieran sintonizar con un televisor en particular. Pero para aquellos que quieran más tendrán que buscar un monitor que acepten entrada de video compuesta y prepararse para hacer un pequeño trabajo de soldadura.

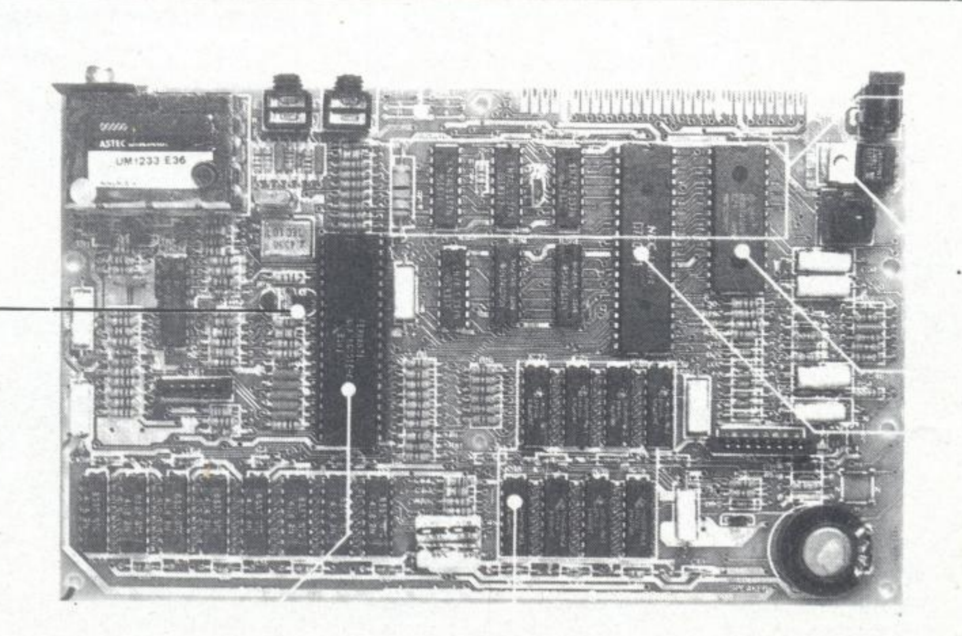
Más abajo del modulador y justo a la izquierda del hueco existente entre TC1 y TC2, hay una pequeña línea marcada con las letras "VID". Al final de uno de los extremos tendremos un pequeño punto de soldadura y al otro extremo un pequeño agujero a través de la placa. No es tan fácil acceder a esta línea puesto que está situada a lo largo de las resistencias situadas en la parte inferior del modulador.

"VID" forma como una pequeña frontera entre el *port* de salida *(port* de expansión) y el modulador. Es obvio que al haber un pequeño hueco en esta línea lo que habrá que hacer es un puente por encima del hueco. Cómo conseguirlo es cosa de cada uno. Una pequeña pieza de alambre bastaría, aunque lo mejor pero mucho más engorroso sería soldar estaño desde el punto de soldadura hasta el agujero. De cualquier forma, una vez conseguido tendrá una señal de video en el *port*.

Solamente son importantes dos líneas en el *port*, que son la de video y la de 0 voltios. Estas están situadas juntas en la parte inferior del *port*. En la entrada, donde se conectará el monitor, la línea de video va hacia la patilla central mientras que la línea de 0 voltios va conectada a la carcasa. Si se quiere se puede adaptar el cable de UHF que viene con el equipo. De cualquier forma cualquier tienda de electricidad tendrá lo que se necesita.

Todo lo que queda ahora es cómo coger la señal de video del *port*. La manera más flexible sería con un conector y un bus de expansión. Por otro lado también se pueden conseguir las señales de otro *port* como por ejemplo si tiene conectada la impresora. De todos modos hay que tener cuidado con algunos programas puesto que se puede causar un volcado de la pantalla por impresora si se mantiene mucho tiempo conectada. Para aquellos que vayan más lejos como para utilizar el monitor en el Spectrum, lo mejor será comprar el conector debido para ello.

VERSION 3



El control del color y el cable plano del Teletext se sustituyeron en el Mark 3 por una utilidad dentro de la ULA para el autoajuste del color. Esto fue un paso importante ya que en los modelos Mark 1 y 2 por mucho que se intentara ajustar la pantalla, en televisiones alemanas y japonesas se convertía en un problema.

COMENTARIOS

Programa: Pheenix.
Tipo: Juego.
Distribuidor: Ventamatic.
Formato: Cinta de cassette.
ZX Spectrum 16 ó 48 K.

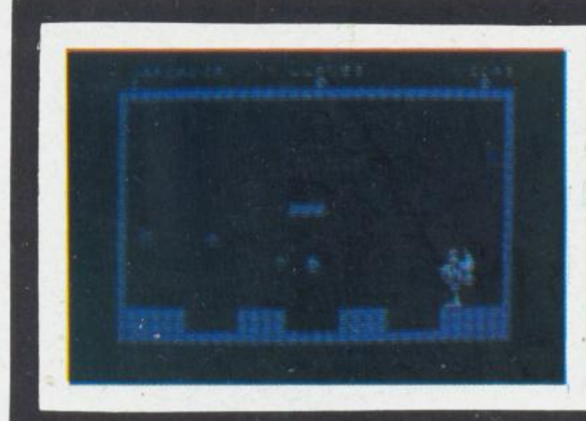
Este juego de tipo Arcade es similar a uno del mismo nombre que abundaba en los bares no hace muchos años. Nosotros controlamos una nave espacial situada en la nave inferior de la pantalla que debe luchar contra cuatro tipos de invasores distintos antes de poder destruir la nave ma-. Estos enemigos se dividen en dos clases. Las dos primeras pantallas son de marcianos normales (con variaciones entre ellos de color y forma) y la tercera y cuarta están formadas por unos huevos que posteriormente se convierten en grandes pájaros. Después de estas cuatro oleadas nos aparece una nave de gran tamaño escoltada por otra escuadrilla de marcianos.

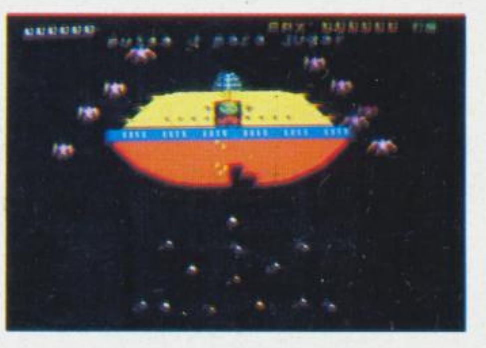
El juego se carga sin dificultad siguiendo las instrucciones de la cinta y una vez cargado se nos da a elegir entre jugar con el teclado o con un joystick de tipo Kempston. A continuación, se nos pide el nivel de dificultad, que está comprendido entre uno y cinco. La calibración de la dificultad está muy bien hecha y mientras el uno es lo bastante lento como para principiantes, el cinco es para jugadores muy, muy rápidos.

Cuando empezamos a jugar suena una música demasiado larga para nuestro gusto y que no se puede parar, las dos primeras veces que se oye es muy simpática pero a la tercera empieza a hartarnos. Después viene la primera oleada con los marcianos que empiezan a aparecer por los bordes izquierdo y derecho de la pantalla y se van colocando en su posición de combate, mientras se están colocando no disparan por lo que resulta muy fácil destruirles. Cuando están colocados todos empiezan a moverse y disparar, la cantidad de tiros que realizan resulta excesiva, incluso en el nivel uno, por lo que resulta extremadamente difícil pasar por debajo de ellos.

Una vez destruida la primera oleada vendrá una segunda del mismo tipo, aunque los marcianos sean del mismo color. Después de ésta aparecen unos huevos de pájaro que se desplazan por la pantalla lateralmente hasta que al cabo de algunos segundos nace el pájaro. Podemos intentar disparar a los huevos y a los pájaros, y debemos intentar destruirlos cuando todavía son huevos, ya que no disparan y son más vulnerables.
Después de dos oleadas de pájaros aparece la nave madre. Es una nave de gran tamaño que ocupa casi toda la pantalla y en la que está el jefe de los invasores. Para destruirle y ganar debemos agujerear el grueso casco de la nave y realizar un agujero en la cinta corrediza que rodea el perímetro del platillo volante, una vez hemos logrado esto hay que disparar a través de los agujeros y destruir al jefe enemigo (que por cierto parece un gato meneando la cola).

El juego está realizado con gran animación incluyendo efectos de sonido. Las distintas pantallas están bien realizadas y los niveles de juego tienen velocidades correctas. Los dos inconvenientes que hemos visto consisten en que al ponernos un escudo protector antibalas no podemos disparar pero si movernos (justo al revés que su homólogo de los bares) y que sus naves disparan demasiado.





PUNTUACION:
ADICCION: 5.
PRESENTACION: 7.
GRAFICOS: 9.
ACCION: 4.

Programa: Polinomios.
Tipo: Educativo.
Distribuidor: ABC SOFT.
Formato: Cinta de cassette.
Spectrum 48K.

Este programa pretende tratar el estudio de los polinomios desde un nivel de primero de **BUP** hasta primero de carrera. Este espectro es demasiado amplio para poder tratarlo en profundidad y, por tanto, no puede hacer un estudio exhaustivo de todo el tema. Esto no es óbice para que explique bien los temas que trae (hasta un total de 20) y de ejemplos claros.

El programa está planteado con una óptica abierta en la que no se intenta guiar al estudiante a unos temas específicos sino que se le da un menú donde puede elegir el tema a tratar, además todos los cálculos se hacen con un polinomio que previamente le introduce el estudiante y que, puede ser de cualquier dimensión (hasta un máximo de 12) pudiendo trabajar con el mismo polinomio o con cualquiera de sus derivadas.

Entre las distintas opciones que tiene figuran: cálculo de derivadas, cálculo de la integral, cálculo de raíces de diversos tipos (enteras, fraccionarias, reales), máximos y mínimos, dibujo del polinomio y de sus derivados, cálculo de áreas encerradas por el polinomio y cálculo de volúmenes, etc.

La realización del programa es correcta y va acompañado de un manual muy completo y de fácil comprensión. Como fallo se podría indicar la lentitud de algunos cálculos en los que hace esperar algo más de lo acostumbrado para un ordenador (después de todo, los ordenadores son rápidos, ¿no?).

PUNTUACION: UTILIDAD: 7. PRESENTACION: 5. CLARIDAD: 8. RAPIDEZ: 4.

Programa: Artist.
Tipo: Utilidades.
Distribuidor: Dinamic soft.
Formato: Cinta de cassette.
ZX Spectrum 48K.

Este programa le permite crear pantallas gráficas con gran sencillez y comodidad ofreciéndole un amplio conjunto de "herramientas" que simplifican notablemente el trabajo, haciendo que la creación de dibujos espectaculares, similares a los que traen muchos juegos comerciales, estén al alcance de todos.

El programa se carga siguiendo los procedimientos normales y una vez puesto en marcha nos aparece un menú desde donde podemos elegir las distintas modalidades de dibujo, estas se pueden mezclar permitiendo una amplia gama de posibilidades.

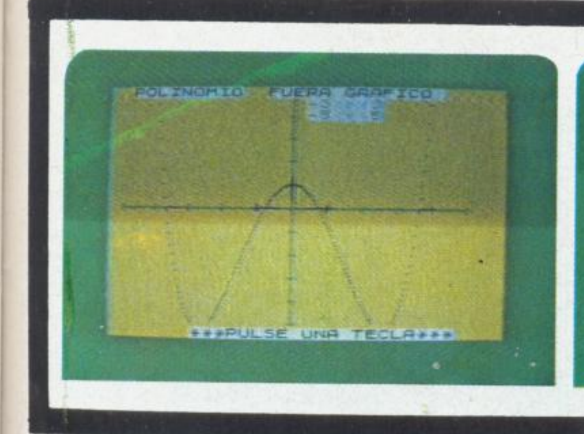
La primera modalidad existente es la de modo *plotter*. Aquí se nos permite colocar puntos individuales, líneas rectas y toda clase de curvas. También tiene una opción destinada a rellenar áreas de un color, con lo que se simplifica enormemente el trabajo. En modo texto permite la introducción de caracteres dentro del dibujo; estos caracteres pueden ser los normales del **Spectrum** o de gran tamaño, especialmente útiles para títulos. Este apartado nos permite un mejor control de los bloques de la pantalla (8 * 8 puntos) pudiendo cambiar el fondo individual de cada uno de los puntos. Otra de las opciones de que dispone el programa es un editor de caracteres *(UDG) que aunque su existencia sea estándar del* **Spectrum**, su creación y manejo no es excesivamente fácil, sobre todo para las personas que no poseen demasiada soltura con la máquina. El programa soluciona este problema, en parte, permitiendo la creación y modificación de estos caracteres, que pueden ser usados dentro del gráfico con el que estemos trabajando o salvados para su uso con otro programa.

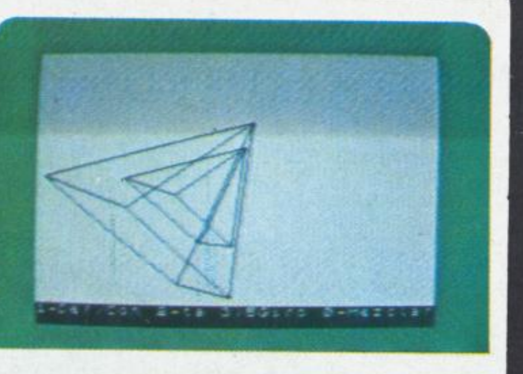
Una curiosa peculiaridad de este programa reside en la posibilidad de trabajar en perspectiva y poder definir figuras tridimensionales, para esto se dispone de un modo especial en el que todos los movimientos se realizan en perspectiva, permitiéndonos, trazar los dibujos en tres dimensiones y añadirlos al gráfico general donde luego se pueden rellenar los lados.

Aparte de todas las opciones indicadas anteriormente dispone de las típicas necesarias para leer y grabar los gráficos en cinta.

El programa está bien desarrollado y funciona a la perfección, aunque se echa a notar en falta una opción que permita la ampliación de una pequeña parte de la pantalla a gran escala para su mejor manejo.

PUNTUACION: UTILIDAD: 7. PRESENTACION: 5. CLARIDAD: 6. RAPIDEZ: 8.





Programa: Túneles marcianos.
Tipo: Juego.
Distribuidor: Ventamatic.
Formato: Cinta de cassette.
Spectrum 48K.

Si hay alguna característica que sea común a la gran mayoría de los juegos, y sobre todo los juegos de acción como este, es la sencillez de sus principios y lo poco que se necesita pensar para poder jugar. Y es justo en este punto en el que el programa Túneles Marcianos llama la atención. Desde un principio hay un objetivo bien definido consistente en coger un cofre y abrirlo para llevarse el tesoro que hay dentro y para poder hacer esto necesitamos siete llaves que hay repartidas por distintas partes del laberinto.

Y si ésta es la presentación típica de un juego de aventuras en el que tendremos que recorrer todos los laberintos mirando cosas, cogiendo, dejando y haciendo otro sinfín de cosas propias de un explorador, la realidad es bastante diferente. Cada habitación del laberinto está llena de monstruos que, como es normal en estos casos, intentan comernos mientras que nosotros con una pistola intentamos matarlos a todos para pasar a la siguiente habitación. El problema es más difícil en las habitaciones que contengan algún objeto interesante (cofre, llaves) ya que este debe ser código antes de matar a todos los monstruos, si no lo hacemoa así, el objeto desaparecerá cuando queden menos de tres monstruos y deberemos volver a empezar a buscarlo.

El juego es fácil de entender y tiene un manejo sencillo por medio de un *joystick* o del teclado. Los gráficos representando a los monstruos y al astronauta son de gran tamaño y de una calidad sorprendente (sobre todo el del astronauta) que le da más realismo al juego. Los niveles de dificultad aumentan de habitación en habitación como era de esperar y resulta bastante difícil pasar de la tercer

Las instrucciones de juego van incluidas (en español) en el mismo programa y todo el juego va acompañado de una serie de tonadas, algunas muy conocidas, que le dan más animación. La presentación es muy sencilla (no tiene ninguna pantalla de portada con un gráfico espectacular) y el manejo por el teclado es muy cómodo y no plantea problemas.

PUNTUACION: ADICCION: 8. PRESENTACION: 5. GRAFICOS: 8. ACCION: 7.

Y porque no sólo de aprender se trata, sino también de divertirse, cerraremos **Todospectrum** con un pequeño repertorio de buenos programas.

Les garantizamos que todos serán probados y verificados, ofreciendo una copia de la representación por pantalla de aquellos aspectos más significativos del programa. Les prometemos también vigilar estrechamente a nuestro maquetista para evitar que "se le vaya la tijera" en los listados de los programas.

Dado que existen dos modelos de Spectrum en cuanto a capacidad de memoria, indicaremos en cada programa la capacidad mínima necesaria. *16K* indicará que corre en un *Spectrum* de *16K* o de capacidades mayores, y *48K* que necesita dicha cantidad de memoria como mínimo, no pudiendo correr en el *Spectrum* de *16K*. Muchos programas utilizarán los gráficos de alta resolución, en cuyo caso lo indicaremos adecuadamente. Veamos un ejemplo:

Barcos

Con un poco de ilusión y grandes dosis de paciencia le auguramos una buena partida al popular juego de los barcos para la próxima navidad. Quizás sea exagerar un poco, a pesar del elevado número de instrucciones a introducir, pero no le exageramos nada al decir que es un estupendo juego con el que podrá jugar contra el ordenador, con una buena resolución gráfica aun no generando ningún carácter gráfico.

Por si acaso es usted nuevo en estas lides de guerra o desea que le refresquemos la memoria, veamos las reglas de juego impuestas por este programa:

1. Dispone de un cuadrado de 14 × 14 donde situar sus navíos.
2. La flota consta de 1 portaviones (4 recuadros), 2 cruceros (3 recuadros), 3 submarinos (2 recuadros) y 5 lanchas (1 recuadro).
3. Las naves no pueden tocarse, debiendo existir al menos un cuadrado libre desde cualquier posición vertical, horizontal o diagonal.
4. Existen dos posibilidades de juego: hundir la flota o jugar a puntos. El primer caso es el clásico juego de los barcos en el que el ganador es el que consigue destruir toda la flota enemiga. La segunda opción consiste en tratar de hundir el mayor número posible de barcos con un número de disparos limitados, acortándose así el juego (especialmente recomendado para nerviosos y ejecutivos atareados).



5. Los disparos son alternos, a excepción de cuando se hace blanco, en cuyo caso sigue jugando el mismo jugador.
6. El ordenador no repite nunca el disparo en un mismo lugar y cuando toca a un barco tiene la costumbre de dispararle aleatoriamente alrededor hasta hundirlo.

Autor: Jorge Turró Noguera. 48K

```
  10 REM Programa "Barcos"
  12 REM Creado por JORDI TURRO
  13 REM ZX SPECTRUM 20-2-1984
  15 CLEAR : CLS : RANDOMIZE
  16 DIM D(14,14): DIM O(14,14)
  17 DIM x(21): DIM y(21)
  18 DIM a(21): DIM l(21)
  19 DIM s(21): DIM t(21)
  20 DIM g(21): DIM h(4)
  25 LET P0=0: LET Po0=0
  26 LET C1=0: LET Co1=0
  27 LET C2=0: LET Co2=0
  28 LET S1=0: LET So1=0
  29 LET S2=0: LET So2=0
  30 LET S3=0: LET So3=0
  31 LET dd=0: LET dj=0
  32 LET AJ=0: LET AO=0
  33 LET JP=0: LET TT=0
  34 LET LL=0: LET To=0
  35 LET Tr=0: LET Ts=0
  36 LET Md=250: LET d=1
  38 PAPER 7
  39 BORDER 5
  40 PRINT INK 1; INVERSE 1;"JUE
GOS DE BARCOS HUNDIR LA FLOTA"
  50 PRINT INK 2; INVERSE 1;AT 1
,6;" REGLAS DEL JUEGO "
  55 PRINT INK 3;"■■■■■■■■■■■■■■
■■■■■■■■■■■■■■■■■■"
  60 PRINT "Primero deberas situ
ar en posicion tus barcos.": PRI
NT
  65 PRINT "El ordenador despues
 se asignarala posicion de los s
uyos.": PRINT
  70 PRINT "La flota estara form
ada por:"
  75 PRINT "1 Portaviones de 4 r
ecuadros"
  80 PRINT "2 Cruceros de 3 recu
adros"
  85 PRINT "3 Submarinos de 2 re
cuadros"
  90 PRINT "5 Lanchas de 1 recua
dro": PRINT
  95 PRINT "Los disparos seran a
lternos , excepto cuando uno hag
a blanco quetirara de nuevo y no
 se contara"
 100 PRINT
 110 PRINT : PRINT INK 1; INVERS
E 1;"Para seguir viendo instrucc
iones pulsa una tecla cualquiera
"
```

Programas

Notas gráficas

ABCDEFGHIJKLMNOPQRSTU
[illegible] GHIJKLMNOPQRSTU

Cada vez que vea un carácter "extraño" (fuera de los disponibles en el Spectrum) bastará con que le busque en esta segunda línea, cambie a modo gráfico (*CAPS SHIFT* + 9) y presione la tecla inmediatamente superior en la primera línea. ¿Fácil no? Como usted ya sabe, con ello aparecerá la letra que pulsa, pero cuando ejecute el programa *(RUN)* se cambiará por el carácter correspondiente generado por el propio programa.

Esperamos que la selección sea de su agrado, y si no lo es díganoslo: ¡admitimos sugerencias!

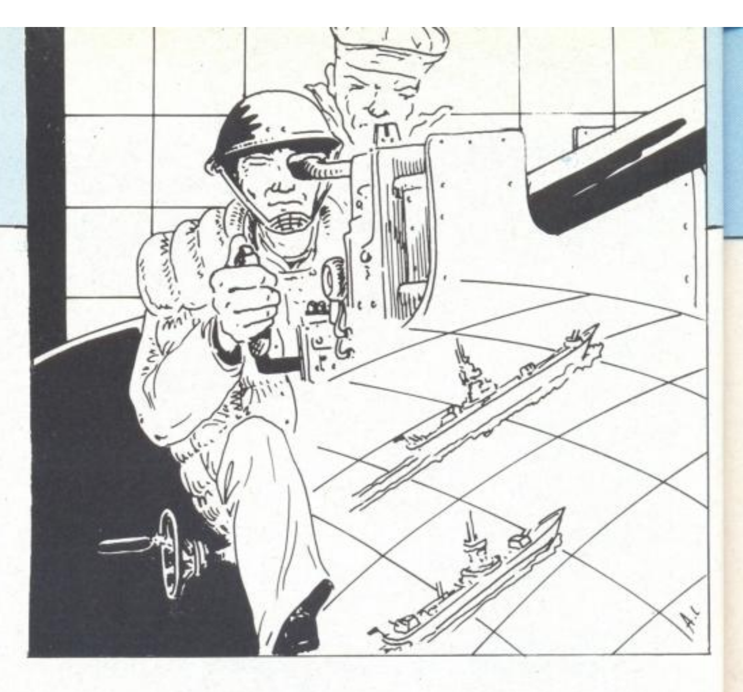
```
 115 PAUSE 0: CLS
 120 PRINT "Puedes jugar a  punt
os o hundir toda la flota"
 125 PRINT : PRINT "Jugando a pu
ntos, maximo 250 disparos en tot
al"
 130 PRINT
 135 PRINT "Portaviones=4 puntos
"
 140 PRINT "Crucero=     6 puntos
"
 145 PRINT "Submarino=  8 puntos
"
 150 PRINT "Lancha=     10 puntos
"
 160 PRINT : PRINT "El tablero i
zquierdo son los disparos del ju
gador y el  derecho del ordenado
r"
 165 PRINT : PRINT "Debajo de el
los saldran los  puntos que vaya
s sumando y en mediolos disparos
 efectuados"
 170 PRINT : PRINT : PRINT INK 1
; INVERSE 1;"Para seguir viendo
instrucciones pulsa una tecla cu
alquiera           "
 175 PAUSE 0: CLS
 180 PRINT "Si juegas a puntos,
el contador de disparos ira rest
ando los disparos que quedan"
 185 PRINT : PRINT "Una vez term
inado el juego a puntos, podras
elegir continuar el mismo"
 190 PRINT : PRINT "Si juegas a
hundir toda la flotael contador
solo  sumara los disparos efectu
ados"
 200 PRINT : PRINT "El ordenador
 al finalizar la partida, te dir
a, donde tenia sus  Barcos"
 210 PRINT : PRINT
 230 PRINT : PRINT : PRINT : PRI
NT INK 2; INVERSE 1;"Para empeza
r el juego pulsa una tecla cualq
uiera                   "
 240 PAUSE 0: CLS
 260 GO SUB 9370

 275 PRINT INK 1; INVERSE 1;AT 1
6,0;"JUGADOR"; INK 2; INVERSE 1;
AT 16,11;"Tiro";AT 16,20; INK 3;
 INVERSE 1;"ORDENADOR"
 280 PRINT INK 1;AT 16,7;":";AJ;
INK 2;AT 16,15;":";TT; INK 3;AT
16,29;":";AO
 285 GO TO 2370
 290 PRINT AT 17,0;"

                              "

 295 PRINT FLASH 1; INK 3;AT 20,
13;"ESPERA"
 297 PRINT INK 2; INVERSE 1;AT 2
1,2;" Me estoy situando los mios
"
 300 REM ASIGNACION POSICION DE
   BARCOS DEL ORDENADOR
 305 REM ASIGNAR 1 PORTAVIONES
 310 LET X=INT (RND*11)
 315 LET Y=INT (RND*11)+1
 320 LET B=INT (RND*10)
 325 IF B>=5 THEN GO TO 380
 335 REM PORTAVIONES EJE X >
 345 FOR n=1 TO 4
 350 LET x(n)=X+n-1
 355 LET y(n)=Y
 360 NEXT n
 370 GO TO 410
 380 REM PORTAVIONES EJE Y v
 385 FOR n=1 TO 4
 390 LET x(n)=X
 395 LET y(n)=Y+n-1
 400 NEXT n
 410 REM ASIGNAR  2 CRUCEROS
 420 FOR n=5 TO 10 STEP 3
 425 LET B=INT (RND*10)
 430 LET G=0
 440 LET X=INT (RND*12)
 445 LET Y=INT (RND*12)+1
 450 FOR p=n TO n+2
 460 IF B>=5 THEN GO TO 490
 470 REM CRUCERO EJE X >
 475 LET x(p)=X+p-n
 480 LET y(p)=Y
 485 GO TO 510
 490 REM CRUCERO EJE Y v
 495 LET x(p)=X
 500 LET y(p)=Y+p-n
 510 GO SUB 840
 520 NEXT p
 530 IF G=1 THEN GO TO 430
 540 NEXT n
 630 REM ASIGNAR 3 SUBMARINOS
 635 FOR n=11 TO 16 STEP 2
 640 LET B=INT (RND*10)
 645 LET G=0
 650 LET X=INT (RND*13)
 655 LET Y=INT (RND*13)+1
 660 FOR p=n TO n+1
 665 IF B>=5 THEN GO TO 690
 670 REM SUBMARINO EJE X >
 675 LET x(p)=X+p-n
 680 LET y(p)=Y
 685 GO TO 705
 690 REM SUBMARINO EJE Y v
 695 LET x(p)=X
 700 LET y(p)=Y+p-n
 705 GO SUB 840
 710 NEXT p
 720 IF G=1 THEN GO TO 645
 730 NEXT n
 770 REM ASIGNAR 5 LANCHAS
 775 FOR n=17 TO 21
 780 LET p=n
 785 LET G=0
 790 LET x(p)=INT (RND*14)
 795 LET y(p)=INT (RND*14)+1
 800 GO SUB 840
 810 IF G=1 THEN GO TO 785
 815 LET g=p
 820 NEXT n
 830 GO TO 920
 840 REM COMPROBAR COINCIDENCIAS
 845 FOR m=n-1 TO 1 STEP -1
```

1



```
 850 IF x(p)=x(m) AND y(p)=y(m)
THEN LET G=1
 855 REM COMPROBAR ESTAN PEGADOS
 860 IF x(p)-1=x(m) AND y(p)=y(m
) THEN LET G=1
 865 IF x(p)+1=x(m) AND y(p)=y(m
) THEN LET G=1
 870 IF x(p)=x(m) AND y(p)-1=y(m
) THEN LET G=1
 875 IF x(p)=x(m) AND y(p)+1=y(m
) THEN LET G=1
 880 REM COMPROBAR CON VERTICES
 885 IF x(p)-1=x(m) AND y(p)-1=y
(m) THEN LET G=1
 890 IF x(p)+1=x(m) AND y(p)-1=y
(m) THEN LET G=1
 895 IF x(p)-1=x(m) AND y(p)+1=y
(m) THEN LET G=1
 900 IF x(p)+1=x(m) AND y(p)+1=y
(m) THEN LET G=1
 905 NEXT m
 910 RETURN
 915 REM PREGUNTAS PARA INICIO
   DEL JUEGO
 920 FLASH 0
 925 GO SUB 7700
 930 BEEP 1.5,14
 940 LET JP=0
 945 IF Md=0 THEN GO SUB 6300: G
O TO 1000
 950 PRINT INK 1;AT 20,0;"A QUE
JUEGAS ?                        "
 955 PRINT INK 2;AT 21,0;"Puntos
=P, o Hundir todo=H             "
 960 INPUT J$
 965 IF J$<>CHR$ 112 AND J$<>CHR
$ 104 THEN GO TO 950
 970 IF J$="h" THEN GO TO 1000
 975 LET JP=1
 980 PRINT INK 1;AT 20,0;"Dime e
l numero maximo disparos "
 985 PRINT INK 2;AT 21,0;"Maximo
 ";Md;" disparos                "
 990 INPUT Nd
 995 IF Nd>Md OR Nd<=0 THEN GO T
O 980
 996 LET Md=Md-Nd
 997 LET TT=Nd
 998 PRINT INK 2;AT 16,16;Nd
1000 REM ASIGNACION JUGADOR
1020 PRINT INK 1;AT 20,0;"Quien
empieza primero ?               "
1025 PRINT INK 2;AT 21,0;"  TU=T
 o YO=Y                         "
1030 INPUT k$
1035 IF k$<>CHR$ 116 AND k$<>CHR
$ 121 THEN GO TO 1020
1040 IF k$="t" THEN GO TO 2970
1150 GO SUB 7720
1151 PRINT INK 1;AT 18,0;"DISPAR
OS JUGADOR                      "
1153 IF TT<100 THEN PRINT AT 16,
18;" "
1154 IF TT<10 THEN PRINT AT 16,1
7;" "
1155 PRINT INK 2;AT 16,16;TT
1157 IF JP=1 THEN GO TO 1170
1160 LET TT=TT+1
1165 GO TO 1180
1170 IF TT=0 THEN GO TO 7800
1175 LET TT=TT-1
1180 REM DISPAROS DE JUGADOR
1182 PRINT INK 1;AT 16,8;AJ
1195 PRINT INK 2;AT 19,0;"Coorde
nadas";AT 20,0;"de disparo"
1198 PRINT INK 1;AT 19,13;"letra
 =";  FLASH 1;AT 19,20;"?": INPUT
 x$: FLASH 0
1200 IF x$<CHR$ 97 OR x$>CHR$ 11
0 THEN GO SUB 4700: GO TO 1198
1203 PRINT INK 2;AT 19,20;x$
1205 PRINT INK 1;AT 20,13;"numer
o=";  FLASH 1;AT 20,20;"?": INPUT
 y: FLASH 0
1210 IF y<1 OR y>14 THEN GO SUB
4700: GO TO 1205
1215 PRINT INK 2;AT 20,20;y
1218 REM COORDENDA (X) DISPARO
  JUGADOR y ASIGNAC. ORDENADOR
1220 IF x$="a" THEN LET s=0
1222 IF x$="b" THEN LET s=1
1224 IF x$="c" THEN LET s=2
1226 IF x$="d" THEN LET s=3
1228 IF x$="e" THEN LET s=4
1230 IF x$="f" THEN LET s=5
1232 IF x$="g" THEN LET s=6
1234 IF x$="h" THEN LET s=7
1236 IF x$="i" THEN LET s=8
1238 IF x$="j" THEN LET s=9
1240 IF x$="k" THEN LET s=10
1242 IF x$="l" THEN LET s=11
1244 IF x$="m" THEN LET s=12
1246 IF x$="n" THEN LET s=13
1250 FOR f=0 TO 13
1255 IF s=f THEN LET x=f*8
1260 NEXT f
1280 GO SUB 1300
1290 GO TO 1330
1300 REM COORDENADA (Y) DISPARO
  y ASIGNAC., JUGADOR y ORDENA.
1302 FOR f=1 TO 14
1305 IF y=f THEN LET w=159-8*(f-
1): LET v=w+8
1310 NEXT f
1320 RETURN
1330 REM ANULAR COINCIDENCIAS
     DISPAROS JUGADOR
1332 IF dj=1 THEN GO TO 1355
1335 LET D(s+1,y)=1
1340 LET dj=1
1350 GO TO 1400
1355 IF D(s+1,y)=1 THEN GO SUB 7
710: GO SUB 4550: GO TO 1180
1360 LET D(s+1,y)=1
1400 GO SUB 8390
2000 REM COMPROBACION ACIERTOS
     DEL JUGADOR de 2005 a 2365
2005 LET A=0: LET c=0
2010 FOR n=1 TO 21
2015 IF s=x(n) AND y=y(n) THEN L
ET A=1: LET c=n: LET l(n)=n
2020 NEXT n
2025 PLOT FLASH 0;x+4,w+4
2027 IF A=0 THEN GO SUB 7300: GO
 TO 3000
2029 REM Acierto Portaviones ?
2030 IF c=1 OR c=2 OR c=3 OR c=4
 THEN GO TO 2080
2034 REM Acierto Cruceros ?
2035 IF c=5 OR c=6 OR c=7 THEN G
O TO 2120
2040 IF c=8 OR c=9 OR c=10 THEN
GO TO 2160
2044 REM Aciertos Submarinos ?
2045 IF c=11 OR c=12 THEN GO TO
2200
```

```
2050 IF c=13 OR c=14 THEN GO TO
2240
2055 IF c=15 OR c=16 THEN GO TO
2280
2060 REM Aciertos lanchas ?
2065 PRINT INK 2;AT y,s;"■"
2067 LET AJ=AJ+10
2070 GO SUB 7600: GO TO 2350
2080 IF l(1)=1 AND l(2)=2 AND l(
3)=3 AND l(4)=4 THEN GO TO 2100
2090 GO SUB 6100: GO SUB 7500: L
ET AJ=AJ+1: GO TO 1180
2100 LET AJ=AJ+1: LET P0=1: LET
JA=1: LET JB=4
2110 GO SUB 6150: GO SUB 7600: G
O TO 2350
2120 IF l(5)=5 AND l(6)=6 AND l(
7)=7 THEN GO TO 2140
2130 GO SUB 6100: GO SUB 7500: L
ET AJ=AJ+2: GO TO 1180
2140 LET AJ=AJ+2: LET C1=1: LET
JA=5: LET JB=7
2150 GO SUB 6150: GO SUB 7600: G
O TO 2350
2160 IF l(8)=8 AND l(9)=9 AND l(
10)=10 THEN GO TO 2180
2170 GO SUB 6100: GO SUB 7500: L
ET AJ=AJ+2: GO TO 1180
2180 LET AJ=AJ+2: LET C2=1: LET
JA=8: LET JB=10
2190 GO SUB 6150: GO SUB 7600: G
O TO 2350
2200 IF l(11)=11 AND l(12)=12 TH
EN GO TO 2220
2210 GO SUB 6100: GO SUB 7500: L
ET AJ=AJ+4: GO TO 1180
2220 LET AJ=AJ+4: LET S1=1: LET
JA=11: LET JB=12
2230 GO SUB 6150: GO SUB 7600: G
O TO 2350
2240 IF l(13)=13 AND l(14)=14 TH
EN GO TO 2260
2250 GO SUB 6100: GO SUB 7500: L
ET AJ=AJ+4: GO TO 1180
2260 LET AJ=AJ+4: LET S2=1: LET
JA=13: LET JB=14
2270 GO SUB 6150: GO SUB 7600: G
O TO 2350
2280 IF l(15)=15 AND l(16)=16 TH
EN GO TO 2300
2290 GO SUB 6100: GO SUB 7500: L
ET AJ=AJ+4: GO TO 1180
2300 LET AJ=AJ+4: LET S3=1: LET
JA=15: LET JB=16
2310 GO SUB 6150: GO SUB 7600
2350 REM ? TODOS HUNDIDOS
2360 IF P0=1 AND C1=1 AND C2=1 A
ND S1=1 AND S2=1 AND S3=1 AND l(
17)=17 AND l(18)=18 AND l(19)=19
 AND l(20)=20 AND l(21)=21 THEN
LET JP=0: GO TO 9880
2365 GO TO 1180
2370 REM ASIGNACION BARCOS DEL
     JUGADOR de 2375 a 2880
2375 PRINT INK 2;AT 17,0;"ASIGNA
CION BARCOS JUGADOR"
2380 PRINT INK 1;AT 18,0;"1 Port
aviones de 4 recuadros"
2390 GO SUB 4300
2400 IF w$="v" THEN GO TO 2460
2405 IF x$>CHR$ 107 THEN GO SUB
4700: GO SUB 4750: GO TO 2370
2407 PLOT FLASH 0;xx+4,w+2
2410 FOR n=0 TO 24 STEP 8
2415 CIRCLE INK 1;xx+4+n,w+4,2
2417 CIRCLE INK 1;xx+3.5+n,w+4,1
2420 NEXT n
2430 FOR n=1 TO 4
2435 LET s(n)=ss+n-1
2440 LET t(n)=y
2445 NEXT n
2450 GO TO 2500
2460 IF y>11 THEN GO SUB 4700: G
O SUB 4750: GO TO 2370
2462 PLOT FLASH 0;xx+4,w+2
2465 FOR n=0 TO 24 STEP 8
2470 CIRCLE INK 1;xx+4,w+4-n,2
2472 CIRCLE INK 1;xx+3.5,w+4-n,1
2475 NEXT n
2480 FOR n=1 TO 4
2485 LET s(n)=ss
2490 LET t(n)=y+n-1
2495 NEXT n
2500 REM ASIGNACION 2 CRUCEROS
2510 PRINT INK 1;AT 18,0;"2 Cruc
eros de 3 recuadros "
2515 FOR n=5 TO 10 STEP 3
2525 LET GG=0
2530 GO SUB 4300
2535 IF w$="v" THEN GO TO 2600
2540 IF ss>28 THEN GO SUB 4700:
GO SUB 4750: GO TO 2525
2545 FOR p=n TO n+2
2550 LET s(p)=ss+p-n
2555 LET t(p)=y
2560 GO SUB 4900
2565 NEXT p
2570 IF GG=1 THEN GO SUB 4550: G
O SUB 4750: GO SUB 4200: GO TO 2
525
2572 PLOT FLASH 0;xx+4,w+2
2575 FOR m=0 TO 16 STEP 8
2580 CIRCLE INK 1;xx+4+m,w+4,2
2582 CIRCLE INK 1;xx+3.5+m,w+4,1
2585 NEXT m
2590 GO TO 2665
2600 IF y>12 THEN GO SUB 4700: G
O SUB 4750: GO TO 2525
2605 FOR p=n TO n+2
2610 LET s(p)=ss
2615 LET t(p)=y+p-n
2620 GO SUB 4900
2625 NEXT p
2630 IF GG=1 THEN GO SUB 4550: G
O SUB 4750: GO SUB 4200: GO TO 2
525
2632 PLOT FLASH 0;xx+4,w+2
2635 FOR m=0 TO 16 STEP 8
2640 CIRCLE INK 1;xx+4,w+4-m,2
2642 CIRCLE INK 1;xx+3.5,w+4-m,1
2650 NEXT m
2665 NEXT n
2680 REM ASIGNACION 3 SUBMARINOS
2685 PRINT INK 1;AT 18,0;"3 Subm
arinos de 2 recuadros "
2690 FOR n=11 TO 16 STEP 2
2695 LET GG=0
2700 GO SUB 4300
2705 IF w$="v" THEN GO TO 2760
2710 IF ss>29 THEN GO SUB 4700:
GO SUB 4750: GO TO 2695
2715 FOR p=n TO n+1
2720 LET s(p)=ss+p-n
2725 LET t(p)=y
2730 GO SUB 4900
2735 NEXT p
2745 IF GG=1 THEN GO SUB 4550: G
O SUB 4750: GO SUB 4250: GO TO 2
695
2746 PLOT FLASH 0;xx+4,w+2
2747 FOR m=0 TO 8 STEP 8
2750 CIRCLE INK 1;xx+4+m,w+4,2
2751 CIRCLE INK 1;xx+3.5+m,w+4,1
2753 NEXT m
2755 GO TO 2810
2760 IF y>13 THEN GO SUB 4700: G
O SUB 4750: GO TO 2695
2765 FOR p=n TO n+2
2770 LET s(p)=ss
2775 LET t(p)=y+p-n
2780 GO SUB 4900
2785 NEXT p
2795 IF GG=1 THEN GO SUB 4550: G
O SUB 4750: GO SUB 4250: GO SUB
2695
2796 PLOT FLASH 0;xx+4,w+2
```

```
2797 FOR m=0 TO 8 STEP 8
2800 CIRCLE INK 1;xx+4,w+4-m,2
2801 CIRCLE INK 1;xx+3.5,w+4-m,1
2803 NEXT m
2810 NEXT n
2820 REM ASIGNACION 5 LANCHAS
2825 PRINT INK 1;AT 18,0;"5 Lanc
has de 1 recuadro"
2827 LET LL=1
2830 FOR n=17 TO 21
2835 LET GG=0
2840 GO SUB 4300
2845 LET p=n
2850 LET s(p)=ss
2855 LET t(p)=y
2860 GO SUB 4900
2870 IF GG=1 THEN GO SUB 4550: G
O SUB 4750: GO TO 2835
2871 PLOT FLASH 0;xx+4,w+2
2872 CIRCLE INK 1;xx+4,w+4,2
2873 CIRCLE INK 1;xx+3.5,w+4,1
2880 NEXT n
2900 GO TO 290
2970 GO SUB 7700
3000 REM DISPAROS ORDENADOR
3002 IF TT<100 THEN PRINT AT 16,
18;" "
3003 IF TT<10 THEN PRINT AT 16,1
7;"  "
3004 PRINT INK 2;AT 16,16;TT;AT
16,30; INK 3;AO
3006 IF JP=1 THEN GO TO 3010
3007 LET TT=TT+1
3008 GO TO 3016
3010 IF TT=0 THEN GO TO 7800
3013 LET TT=TT-1
3016 PRINT INK 3;AT 18,0;"DISPAR
OS ORDENADOR"
3020 PRINT FLASH 1; INK 3;AT 20,
13;"ESPERA"
3025 IF Ts>=2 THEN GO TO 5910
3030 IF To<>0 THEN GO TO 5830
3035 LET ss=INT (RND*14)+17
3040 LET y=INT (RND*14)+1
3042 LET ssr=ss: LET yr=y
3045 IF dd=1 THEN GO TO 3055
3047 LET dd=1
3050 LET O(ss-16,y)=1
3052 GO TO 3060
3055 IF O(ss-16,y)=1 THEN GO TO
3025
3057 LET O(ss-16,y)=1
3060 GO SUB 5670
3070 PLOT FLASH 1; INK 2;xx+4,w+
4
3075 PLOT xx+2,w+2: DRAW INK 2;4
,4
3080 PLOT xx+2,v-2: DRAW INK 2;4
,-4
3085 PRINT INK 3;AT 20,0;"Mi dis
paro ya esta efectuado"
3090 PRINT INK 2;AT 21,0;"Para c
ontinuar pulse una tecla"
3095 PAUSE 0: PLOT FLASH 0;xx+4,
w+4
3100 REM COMPR.ACIERTO ORDENADOR
3110 LET Aa=0: LET Cc=0
3115 FOR n=1 TO 21
3120 IF ss=s(n) AND y=t(n) THEN
GO TO 3122
3121 GO TO 3126
3122 LET Aa=1
3123 LET Cc=n
3124 LET g(n)=n
3126 NEXT n
3128 IF Aa=1 THEN GO TO 3140
3130 GO SUB 7300
3138 GO TO 1150
3140 REM Acierto Portaviones ?
3145 IF Cc=1 OR Cc=2 OR Cc=3 OR
Cc=4 THEN GO TO 3240
3150 REM Acierto Cruceros ?
3160 IF Cc=5 OR Cc=6 OR Cc=7 THE
N GO TO 3280
3165 IF Cc=8 OR Cc=9 OR Cc=10 TH
EN GO TO 3320
3170 REM Acierto Submarinos c
3180 IF Cc=11 OR Cc=12 THEN GO T
O 3360
3185 IF Cc=13 OR Cc=14 THEN GO T
O 3400
3190 IF Cc=15 OR Cc=16 THEN GO T
O 3440
3200 REM Aciertos Lanchas
3210 PRINT INK 2;AT y,ss;"■"
3215 LET AO=AO+10
3217 LET JA=Cc: LET JB=Cc
3220 GO SUB 7600
3225 GO SUB 7900
3230 GO TO 3500
3240 IF Po0=1 THEN GO TO 3280
3245 IF g(1)=1 AND g(2)=2 AND g(
3)=3 AND g(4)=4 THEN GO TO 3260
3250 GO SUB 6200: GO SUB 7500: L
ET AO=AO+1: GO TO 5715
3260 LET AO=AO+1: LET Po0=1
3265 LET JA=1: LET JB=4
3270 GO SUB 6250: GO SUB 7600: G
O SUB 7900: GO TO 3500
3280 IF Co1=1 THEN GO TO 3320
3285 IF g(5)=5 AND g(6)=6 AND g(
7)=7 THEN GO TO 3300
3290 GO SUB 6200: GO SUB 7500: L
ET AO=AO+2: GO TO 5715
3300 LET AO=AO+2: LET Co1=1
3305 LET JA=5: LET JB=7
3310 GO SUB 6250: GO SUB 7600: G
O SUB 7900: GO TO 3500
3320 IF Co2=1 THEN GO TO 3360
3325 IF g(8)=8 AND g(9)=9 AND g(
10)=10 THEN GO TO 3340
3330 GO SUB 6200: GO SUB 7500: L
ET AO=AO+2: GO TO 5715
3340 LET AO=AO+2: LET Co2=1
3345 LET JA=8: LET JB=10
3350 GO SUB 6250: GO SUB 7600: G
O SUB 7900: GO TO 3500
3360 IF So1=1 THEN GO TO 3400
3365 IF g(11)=11 AND g(12)=12 TH
EN GO TO 3380
3370 GO SUB 6200: GO SUB 7500: L
ET AO=AO+4: GO TO 5715
3380 LET AO=AO+4: LET So1=1
3385 LET JA=11: LET JB=12
3390 GO SUB 6250: GO SUB 7600: G
O SUB 7900: GO TO 3500
3400 IF So2=1 THEN GO TO 3440
3405 IF g(13)=13 AND g(14)=14 TH
EN GO TO 3420
3410 GO SUB 6200: GO SUB 7500: L
ET AO=AO+4: GO TO 5715
3420 LET AO=AO+4: LET So2=1
3425 LET JA=13: LET JB=14
3430 GO SUB 6250: GO SUB 7600: G
O SUB 7900: GO TO 3500
3440 IF So3=1 THEN GO TO 3480
3445 IF g(15)=15 AND g(16)=16 TH
EN GO TO 3460
3450 GO SUB 6200: GO SUB 7500: L
ET AO=AO+4: GO TO 5715
3460 LET AO=AO+4: LET So3=1
3465 LET JA=15: LET JB=16
3470 GO SUB 6250: GO SUB 7600: G
O SUB 7900: GO TO 3500
3480 REM COMPROBAR TOTAL ACIERTO
3500 IF Po0=1 AND Co1=1 AND Co2=
1 AND So1=1 AND So2=1 AND So3=1
AND g(17)=17 AND g(18)=18 AND g(
19)=19 AND g(20)=20 AND g(21)=21
 THEN LET JP=0: GO TO 9915
3510 GO TO 3000
4200 IF GG=0 THEN GO TO 4230
4205 FOR u=n TO n+2
4210 LET s(u)=0
4215 LET t(u)=0
4220 NEXT u
```

```
4230 RETURN
4250 IF GG=0 THEN GO TO 4280
4255 FOR u=n TO n+1
4260 LET s(u)=0
4265 LET t(u)=0
4270 NEXT u
4280 RETURN
4300 REM SUBRUTINA PARCIAL ASIG
     NACION JUGADOR
4310 LET GG=0
4320 GO SUB 4400
4325 GO SUB 4500
4330 GO SUB 5600
4332 GO SUB 4850
4335 IF GG=1 THEN GO SUB 4550: G
O TO 4300
4337 GO SUB 4450
4340 PLOT FLASH 1; INK 1;xx+4,w+
2
4345 GO SUB 4800
4350 IF q$="n" THEN GO TO 4360
4355 GO TO 4375
4360 CIRCLE OVER 1;xx+4,w+4,2
4365 PLOT FLASH 0;xx+4,w+2
4367 PLOT OVER 1;xx+4,w+2
4370 GO TO 4320
4375 IF LL=1 THEN RETURN
4380 GO SUB 4650
4385 RETURN
4400 REM COORDENADA x$ POSICION
     BARCOS JUGADOR
4405 FLASH 1
4410 PRINT INK 1;AT 21,0;"Letra
? "
4415 INPUT x$
4420 IF x$<CHR$ 97 OR x$>CHR$ 11
0 THEN GO SUB 4700: GO TO 4405
4425 FLASH 0
4430 GO SUB 4600
4435 RETURN
4450 REM INDICAR COORDENADA
4455 CIRCLE INK 1;xx+4,w+4,2
4465 RETURN
4500 REM COORDENADA y POSICION
     BARCOS JUGADOR
4510 PRINT FLASH 1; INK 1;AT 21,
0;"Numero ? "
4515 INPUT y
4520 IF y<1 OR y>14 THEN GO SUB
4700: GO TO 4505
4525 FLASH 0
4530 PRINT AT 21,0;"Numero ";y
4535 RETURN
4550 REM INDICAR COINCIDENCIA
4555 PRINT AT 20,0;"           "
4560 BRIGHT 1
4565 PRINT AT 21,0;"Coincidencia
"
```



```
4570 BEEP 1,0: BRIGHT 0
4575 PRINT AT 21,0;"
"
4580 RETURN
4600 REM REPRES. LETRA ASIGNADA
4605 PRINT AT 20,0;"Letra   ";x$
4610 RETURN
4650 REM ? HORIZONTAL/VERTICAL
4655 PRINT INK 1;AT 20,0;"Como l
o deseas ?
4660 PRINT INK 2;AT 21,0;"Horizo
ntal=H o Vertical=V "
4665 INPUT w$
4670 IF w$<>CHR$ 118 AND w$<>CHR
$ 104 THEN GO TO 4655

4675 PRINT AT 20,0;"

4680 PRINT AT 21,0;"

4685 RETURN
4700 REM INDICAR FUERA DE RANGO
4705 FLASH 0: BRIGHT 1
4710 PRINT AT 21,0;"Fuera de ran
go"
4715 BEEP 1,0: BRIGHT 0
4720 PRINT AT 21,0;"
"
4730 RETURN
4750 REM BORRAR COORDENADA
4755 CIRCLE OVER 1;xx+4,w+4,2
4760 PLOT FLASH 0;xx+4,w+2
4765 PLOT OVER 1;xx+4,w+2
4770 RETURN
4800 REM ? CONFORME POSICION   A
SIGNACION JUGADOR
4805 PRINT AT 20,0;"

4810 PRINT INK 1;AT 21,0;"Esta c
onforme ? No=N Si=S "
4815 INPUT q$
4817 IF q$<>CHR$ 115 AND q$<>CHR
$ 110 THEN GO TO 4800
4820 PRINT AT 21,0;"

4825 RETURN
4850 REM COMPROBAR COINCIDENCIA
     INDICACION COORDENADA
4855 FOR i=1 TO 20
4860 IF ss=s(i) AND y=t(i) THEN
LET GG=1
4865 NEXT i
4870 RETURN
4900 REM COMROBAR COINCIDENCIAS
4902 PRINT FLASH 1; INK 3;AT 20,
13;"ESPERA"
4905 FOR m=n-1 TO 1 STEP -1
4910 IF s(p)=s(m) AND t(p)=t(m)
THEN LET GG=1
4920 REM ESTAN PEGADOS ?
4922 IF s(p)-1=s(m) AND t(p)=t(m
) THEN LET GG=1
4925 IF s(p)+1=s(m) AND t(p)=t(m
) THEN LET GG=1
4930 IF s(p)=s(m) AND t(p)-1=t(m
) THEN LET GG=1
4935 IF s(p)=s(m) AND t(p)+1=t(m
) THEN LET GG=1
4940 REM SE TOCAN DE VERTICES ?
4945 IF s(p)-1=s(m) AND t(p)-1=t
(m) THEN LET GG=1
4950 IF s(p)-1=s(m) AND t(p)+1=t
(m) THEN LET GG=1
4955 IF s(p)+1=s(m) AND t(p)+1=t
(m) THEN LET GG=1
4960 IF s(p)+1=s(m) AND t(p)-1=t
(m) THEN LET GG=1
4985 NEXT m
4988 PRINT FLASH 0;AT 20,13;"
"
4990 RETURN
5590 REM COORDENADA (X) ASIGNAC.
 JUGADOR Y DISPARO ORDENADOR
```

```
5600 IF x$="a" THEN LET ss=17
5605 IF x$="b" THEN LET ss=18
5610 IF x$="c" THEN LET ss=19
5615 IF x$="d" THEN LET ss=20
5620 IF x$="e" THEN LET ss=21
5625 IF x$="f" THEN LET ss=22
5630 IF x$="g" THEN LET ss=23
5635 IF x$="h" THEN LET ss=24
5640 IF x$="i" THEN LET ss=25
5645 IF x$="j" THEN LET ss=26
5650 IF x$="k" THEN LET ss=27
5655 IF x$="l" THEN LET ss=28
5660 IF x$="m" THEN LET ss=29
5665 IF x$="n" THEN LET ss=30
5670 FOR d=17 TO 30
5675 LET l=d-17
5680 IF ss=d THEN LET xx=136+l*8
5685 NEXT d
5700 GO SUB 1300
5710 RETURN
5715 REM PROBAR TOCADOS BARCOS
  DE JUGADOR POR EL ORDENADOR
5716 PRINT INK 3;AT 16,30;AO
5720 LET Ts=Ts+1
5725 IF To<>0 THEN GO TO 5880
5730 LET Cd=INT (RND*8)+1
5735 IF Cd=1 THEN RESTORE 5850
5737 IF Cd=2 THEN RESTORE 5852
5740 IF Cd=3 THEN RESTORE 5854
5742 IF Cd=4 THEN RESTORE 5856
5744 IF Cd=5 THEN RESTORE 5858
5746 IF Cd=6 THEN RESTORE 5860
5748 IF Cd=7 THEN RESTORE 5862
5750 IF Cd=8 THEN RESTORE 5864
5770 FOR n=1 TO 4
5775 READ D
5780 LET h(n)=D
5785 NEXT n
5790 RESTORE
5825 LET Tr=1
5830 LET To=h(Tr)
5835 LET ss=ssr: LET y=yr
5840 LET Tr=Tr+1
5850 DATA 1,2,3,4
5852 DATA 2,1,4,3
5854 DATA 3,2,1,4
5856 DATA 4,1,3,2
5858 DATA 1,3,4,2
5860 DATA 3,4,2,1
5862 DATA 4,3,2,1
5864 DATA 2,3,4,1
5880 REM SEGUIR EN MISMA LINEA
5885 IF To=1 THEN GO TO 5950
5890 IF To=2 THEN GO TO 5980
5895 IF To=3 THEN GO TO 6010
5900 IF To=4 THEN GO TO 6040
5910 REM PROBAR AL REVES
5915 LET ss=ssr: LET y=yr
5920 IF To=1 THEN LET To=2: GO T
O 5880
5925 IF To=2 THEN LET To=1: GO T
O 5880
5930 IF To=3 THEN LET To=4: GO T
O 5880
5935 IF To=4 THEN LET To=3: GO T
O 5880
5950 REM PROBAR TOCADO ARRIBA
5955 IF y<=1 THEN GO TO 5830
5960 LET y=y-1
5970 GO TO 3055
5980 REM PROBAR TOCADO ABAJO
5985 IF y>=14 THEN GO TO 5830
5990 LET y=y+1
6000 GO TO 3055
6010 REM PROBAR TOCADO IZQUIERDA
6015 IF ss<=17 THEN GO TO 5830
6020 LET ss=ss-1
6030 GO TO 3055
6040 REM PROBAR TOCADO DERECHA
6045 IF ss>=30 THEN GO TO 5830
6050 LET ss=ss+1
6060 GO TO 3055
6100 REM INDICAR TOCADO JUGADOR
6110 PLOT x+4,w+1: DRAW INK 2;0,
6
6120 PLOT x+1,w+4: DRAW INK 2;6,
0
6125 CIRCLE INK 2;x+4,w+4,2
6130 RETURN
6150 REM INDICAR HUNDIDO JUGAD.
6160 FOR n=JA TO JB
6170 PRINT INK 2;AT y(n),x(n);"■
"
6180 NEXT n
6190 RETURN
6200 REM INDI. TOCADO ORDENADOR
6210 PLOT xx+4,w+1: DRAW INK 2;0
,6
6220 PLOT xx+1,w+4: DRAW INK 2;6
,0
6230 RETURN
6250 REM INDI. HUNDIDO ORDENAD.
6254 LET To=0
6256 LET Tr=0: LET Ts=0
6260 FOR n=JA TO JB
6270 PRINT INK 2;AT t(n),s(n);"■
"
6280 NEXT n
```

```
Para seguir viendo instrucciones
 pulsa una tecla cualquiera

Puedes jugar a  puntos o hundir
toda la flota

Jugando a puntos, maximo 250 dis
paros en total

Portaviones=4 puntos
Crucero=    6 puntos
Submarino=  8 puntos
Lancha=    10 puntos

El tablero izquierdo son los dis
paros del jugador y el  derecho
del ordenador

Debajo de ellos saldran los  pun
tos que vayas sumando y en medio
los disparos efectuados
```

```
6290 RETURN
6300 REM MINIMO JUEGO A PUNTOS
6305 GO SUB 7700
6310 PRINT FLASH 1; INK 2;AT 19,
6;"FIN JUEGO A PUNTOS";AT 20,3;"
DESDE AHORA HUNDIR TODOS"
6320 PRINT INK 1;AT 21,0;"Para c
ontinuar pulsa una tecla "
6330 PAUSE 0
6335 FLASH 0
6340 GO SUB 7700
6345 RETURN
7300 REM INDICAR AGUA
7310 GO SUB 7700
7325 BRIGHT 1
7330 PRINT AT 20,11;"  AGUA  "
7335 PRINT AT 21,11;"========"
7340 BEEP 1,-2: BRIGHT 0
7350 GO SUB 7750
7360 RETURN
7500 REM INDICAR TOCADO
7510 GO SUB 7700
7515 BRIGHT 1
7520 PRINT AT 20,12;" TOCADO "
7525 PRINT AT 21,12;"========"
7530 FOR T=1 TO 3
7535 BEEP .3,T: BEEP .3,0
```

```
7540 NEXT T
7550 BRIGHT 0
7555 GO SUB 7750
7560 RETURN
7600 REM INDICAR HUNDIDO
7610 GO SUB 7700
7620 BRIGHT 1
7625 PRINT AT 20,12;" HUNDIDO "
7630 PRINT AT 21,12;"========="
7635 FOR T=1 TO 4
7640 BEEP .2,14: BEEP .2,1
7645 NEXT T
7650 BRIGHT 0
7655 GO SUB 7750
7660 RETURN
7690 REM BORRAR RESTO IMPRESO
7700 PRINT AT 18,0;"
                                "
7710 PRINT AT 19,0;"
                                "
7720 PRINT AT 20,0;"
                                "
7730 PRINT AT 21,0;"
                                "
7740 RETURN
```

```
7750 PRINT AT 20,11;"           "
7760 PRINT AT 21,11;"           "
7770 RETURN
7800 REM FIN PARTIDA POR PUNTOS
7805 GO SUB 7700
7810 IF AO=AJ THEN GO TO 9850
7820 IF AJ<AO THEN GO TO 9915
7830 GO TO 9880
7850 REM SONIDO EMPATE Y GANADOR
7855 FOR n=1 TO 3
7860 BEEP .1,10: BEEP .1,10: BEE
P .1,15: BEEP .1,10: BEEP .1,6:
BEEP .1,10
7865 NEXT n
7870 RETURN
7900 REM ANULAR DISPAROS DE ORDE
  NADOR ALREDEDOR HUNDIDO JUGAD.
7920 FOR c=JA TO JB
7930 IF s(c)=17 THEN GO TO 7940
7935 LET O(s(c)-17,t(c))=1
7940 IF s(c)=30 THEN GO TO 7950
7945 LET O(s(c)-15,t(c))=1
7950 IF t(c)=1 THEN GO TO 7960
7955 LET O(s(c)-16,t(c)-1)=1
7960 IF t(c)=14 THEN GO TO 7970
```

```
Para seguir viendo instrucciones
  pulsa una tecla cualquiera

Si juegas a puntos, el contador
de disparos ira restando los dis
paros que quedan

Una vez terminado el juego a pun
tos, podras elegir continuar el
mismo

Si juegas a hundir toda la flota
el contador solo  sumara los dis
paros efectuados

El ordenador al finalizar la par
tida, te dira, donde tenia sus
Barcos
```

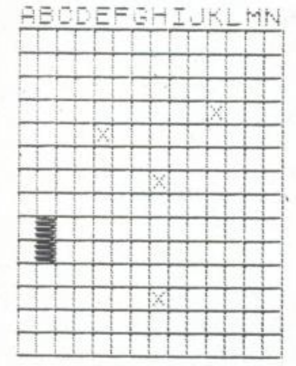
```
JUEGOS DE BARCOS HUNDIR LA FLOTA
       REGLAS DEL JUEGO
================================
Primero deberas situar en posici
on tus barcos.

El ordenador despues se asignara
la posicion de los suyos.

La flota estara formada por:
1 Portaviones de 4 recuadros
2 Cruceros de 3 recuadros
3 Submarinos de 2 recuadros
5 Lanchas de 1 recuadro

Los disparos seran alternos , ex
cepto cuando uno haga blanco que
tirara de nuevo, y no se contara
```

```
Para empezar el juego pulsa una
tecla cualquiera
```




```
JUGADOR: 8    Tiro: 6    ORDENADOR: 2
```

```
7965 LET O(s(c)-16,t(c)+1)=1
7970 IF s(c)=17 OR t(c)=1 THEN G
O TO 7980
7975 LET O(s(c)-17,t(c)-1)=1
7980 IF s(c)=17 OR t(c)=14 THEN
GO TO 7990
7985 LET O(s(c)-17,t(c)+1)=1
7990 IF s(c)=30 OR t(c)=1 THEN G
O TO 8000
7995 LET O(s(c)-15,t(c)-1)=1
8000 IF s(c)=30 OR t(c)=14 THEN
GO TO 8020
8005 LET O(s(c)-15,t(c)+1)=1
8020 NEXT c
8030 RETURN
8390 REM INDICAR DISPARO JUGADOR
8395 PLOT FLASH 1; INK 2;x+4,w+4
8400 PLOT x+2,w+2: DRAW INK 2;4,
4
8410 PLOT x+2,v-2: DRAW INK 2;4,
-4
8412 RETURN
9370 REM FORMACION DE TABLAS
9380 FOR n=0 TO 112 STEP 8
9390 PLOT n,167: DRAW 0,-112
9400 PLOT 0,167-n: DRAW 112,0
9410 NEXT n
```



```
9420 PRINT INK 1;AT 0,0;"ABCDEFG
HIJKLMN   ABCDEFGHIJKLMN"
9430 FOR n=1 TO 14
9440 PRINT INK 1;AT n,15;n
9450 NEXT n
9460 FOR n=136 TO 248 STEP 8
9470 PLOT n,167: DRAW 0,-112
9480 PLOT 136,303-n: DRAW 112,0
9490 NEXT n
9500 RETURN
9850 REM IGUALDAD DE RESULTADOS
9853 PRINT INK 1;AT 16,8;AJ;AT 1
6,30; INK 3;AO
9855 PRINT FLASH 1; INK 1;AT 18,
3;" HEMOS EMPATADO A PUNTOS "
9857 GO SUB 7850
9860 GO TO 9895
9880 REM FLOTA HUNDIDA ORDENAD.
9882 PRINT INK 1;AT 16,8;AJ
9883 GO SUB 7700
9885 PRINT FLASH 1; INK 1;AT 18,
3;" ENHORABUENA ME HAS GANADO "
9887 GO SUB 7850
9890 IF JP=0 THEN GO TO 9940
9895 PRINT INK 2; INVERSE 1;AT 2
0,2;" Continuas la partida ? s/n

9900 INPUT q$
9902 IF q$<>CHR$ 110 AND q$<>CHR
$ 115 THEN GO TO 9895
9904 IF q$="s" THEN GO TO 925
9912 GO TO 9923
9915 PRINT INK 3;AT 16,30;AO
9916 GO SUB 7700
9917 PRINT FLASH 1; INK 2;AT 18,
4;"TE HE GANADO LA PARTIDA"
9918 BEEP .3,-10: BEEP .1,-15: B
EEP .1,-15: BEEP .1,-15: BEEP .3
,-14: BEEP .6,-15: BEEP .3,-11:
BEEP .6,-10
9920 IF JP=1 THEN GO TO 9895
9923 PRINT INK 4; INVERSE 1;AT 2
0,2;"Fijate donde tengo mis Barc
os"
9925 FOR n=1 TO 21
9928 IF l(n)=n THEN GO TO 9935
9930 PRINT INK 3;AT y(n),x(n);"■
"
9935 NEXT n
9940 GO SUB 7730
9950 PRINT INK 3; INVERSE 1;AT 2
1,0;" Deseas combatir de nuevo ?
 s/n "
9960 INPUT Q$
9962 FLASH 0
9965 IF Q$<>CHR$ 115 AND Q$<>CHR
$ 110 THEN GO TO 9940
9980 IF Q$="s" THEN GO TO 10
9985 CLS
9987 FOR n=6 TO 1 STEP -1
9990 PRINT FLASH 1; INK n; INVER
SE 1;AT 11,10;"ADIOS AMIGO"
9993 BEEP .3,16-n
9995 NEXT n
```

SOFTWARE CENTER

sinclair
SEIKOSHA
indescomp
SHARP

★LAS MEJORES MARCAS
★EL MEJOR SERVICIO
★LOS PRECIOS MAS JUSTOS
★LA MAYOR GARANTIA



Y AHORA, ADEMAS ...

ORIC & SPECTRAVIDEO™

↘ VIDEO-GAMES
- CLUB DE VIDEO JUEGOS

↘ GAMES CLUB
- CLUB DE USUARIOS, COMMODORE-64, SPECTRUM Y ORIC

Games Club's

RED NACIONAL DE CLUBS

Aceptamos nuevos grupos federados.
Inmejorables condiciones y asesoramiento.

CONSULTENOS !!!



PONEMOS A SU DISPOSICION «EL CATALOGO» DE SOFTWARE MAS Y MEJOR SURTIDO DEL MERCADO PARA *SPECTRUM*, *COMMODORE* Y *ORIC*.

EL CATALOGO

oferta de verano, solo hasta el 1 de octubre

SHINWA
CP 80 F/T

Impresora matricial 80 columnas con set de caracteres españoles, totalmente compatible.

SHINWA CP80 F/T es la nueva impresora. Con tecnología actual y precio competitivo, ofrece las dos características que hoy día hay que exigir a una buena impresora: fiabilidad y calidad de impresión.

Pero la SHINWA CP80 F/T no se queda ahí: ofrece una resolución de 640 puntos por línea, juego de caracteres españoles y una gran variedad de posibilidades en la impresión de textos: normal, comprimido, doble ancho, super índices subíndices reducidos, etc. La impresora se suministra con interface tipo CENTRONICS. Opcionalmente, se puede conectar un interface RS-232

P.V.OFERTA
55 200 -

DATALEC

P.V. OFERTA
26 320 -



Monitor monocromo para visualización de datos.

El monitor DATALEC, con su pantalla de fósforo verde P-31 de 12 pulgadas, es la pantalla de visualización ideal para presentación de datos y gráficos en alta resolución.

A + L centro de formación. Informática general y BASIC. Prácticas con COMMODORE 64.
INFORMACION: c/ Manso, 17 tel.: 325 87 71

SE BUSCAN¡ los mejores PROGRAMADORES.
Pagamos excelentes royalties. Garantia y seriedad total.

Nombre .... Dirección ....
Población .... Provincia ....
Distrito Postal .... Teléfono ....

BOLETIN DE INFORMACION Tel.: 219 10 90
remitir a
SOFTWARE CENTER
Avda. Mistral 10 „ 1 D izq. BARCELONA
08015

ATENCION!!! A PARTIR DEL 1º DE AGOSTO ENTRA EN FUNCIONAMIENTO EL «MERCA-TELEFONO» MARCANDO EL *219 10 90 (93)* PODRA COMPRAR-CAMBIAR-VENDER CUALQUIER ARTICULO.

Archivo

Si es usted un afortunado que dispone de una variada colección de cintas de programas; si es un poco desordenado y si tiene un microdrive, le ofrecemos la posibilidad de que ponga su vida en orden al menos por lo que al catálogo de programas se refiere.

Con este programa, diseñado especialmente para el almacenamiento de datos en microdrive, puede llevar su registro particular de programas, indicando código, nombre, si es inglesa, e incluso si se trata de una cinta original o copia.

Recuerde no retirar la cinta del microdrive hasta haber finalizado sus operaciones y, en caso de *BREAK* cierre los ficheros *(CLOSE # 4)* para que no se le escape nada (léase para que todo quede grabado).

Por supuesto, le aconsejamos que amplíe el programa para indicar en sus futuros datos el número de todospectrum de donde lo obtuvo.

Autor: Javier García Ariz. 16K.

```
   1 REM JAVIER GARCIA ARIZ -198
4
   5 CLEAR : CLS : BORDER 0: PAP
ER 0: INK 7
  10 CLS : PRINT AT 0,0;"=======
=========================";AT 21
,0;"============================
===="
  15 FOR S=1 TO 20: PRINT AT S,0
;"#";AT S,31;"#": NEXT S
  17 PRINT AT 1,1;"Este es un pr
ograma para        ##almacenar la
lista de cintas  ##que tiene, si
ga todas las       ##instrucciones
 como se indica    ##
                   ##RECUERDE:
                   ##-Tener siempr
e cartucho puesto##-Trabajar en
mayusculas.        ##-Al anadir fi
chas desaparece    ## el fichero a
ntiguo.            ##-Por la impre
sora no aparecen ## or/cop ni si
 es inglesa.       ##-En caso de "
BREAK debe cerrar## los ficheros
 (CLOSE# 4)"
  18 PRINT AT 18,1;"PULSA TECLA"
  19 GO SUB 1000
  20 CLS : PRINT AT 0,0;"=======
=========================";AT 21
,0;"============================
===="
  25 FOR S=1 TO 20: PRINT AT S,0
;"#";AT S,31;"#": NEXT S
  30 FLASH 1: PRINT AT 5,4;"A";A
T 8,4;"B";AT 11,4;"C";AT 14,4;"D
": FLASH 0
  35 PRINT AT 5,6;"CREAR FICHERO
";AT 8,6;"LISTAR FICHERO";AT 11,
6;"IMPRIMIR FICHERO (con rs)";AT
 14,6;"ANADIR FICHAS"
  40 LET V$=INKEY$: IF V$="" THE
N GO TO 40
  45 IF V$="A" THEN GO TO 80
```

Boxeo

Aunque esté en baja forma o incluso aunque no sea partidario del boxeo, nos permitimos sugerirle que elija contrincante, se meta en el cuadrilátero y perfeccione su gancho. Al principio su oponente le hará besar la lona dejándole *KO* con gran facilidad, pero es sólo cuestión de tiempo.

Dispone de tres asaltos para machacar a su oponente y ambos púgiles son ambidiestros, es decir, preste atención tanto a su izquierda como a su derecha. El árbitro es muy severo y no le admitirá golpes bajos.

La resolución gráfica es muy buena, no faltando la chica que anuncia —cartel en mano— el número de asalto en el descanso. Buenas gráficas, buen juego y buen sistema para desahogarse.

Notas gráficas

ABCDEFGHIJKLMNOPQRSTU

[graphic]U

Autor: P. Stanley 16K

```
   1 BORDER 5: PAPER 5: INK 0: C
LS
   2 REM @ P.STANLEY
   5 LET s1=0: LET s2=s1
   8 GO SUB 7000
  10 GO SUB 9000
  15 LET round=1
  20 GO SUB 9500
  30 GO SUB 9550
  62 PRINT AT 17,2;round
  63 BEEP .25,0: BEEP .25,0
  65 LET y=2: LET x=7: LET y1=y:
 LET x1=x
  70 LET g=23: LET f=17: LET g1=
g: LET f1=f
  90 LET tm=3: LET ts=2
  95 LET e=0
 100 PRINT AT y1,x1;"  ";AT y1+1
,x1;"  ";AT y1+2,x1;"  "; INK 2;A
T y,x;"█"; PAPER 0;AT y+1,x;".:"
; PAPER 6;AT y+2,x;"█": LET y1=
y: LET x1=x
 120 PRINT AT f1,g1;"  ";AT f1+1
,g1+1;" ";AT f1+2,g1;"  ";AT f,g
;"█"; PAPER 7;AT f+1,g+1;".:"; P
```



2

```
  50 IF V$="B" THEN GO TO 400
  55 IF V$="C" THEN GO TO 600
  60 IF V$="D" THEN GO TO 800
  75 GO TO 40
  79 STOP : REM $$$$$$$$$$$$$$$$
$$$$$$
  80 CLS : INPUT "NOMBRE DE EL F
ICHERO?:";P$: OPEN #4;"M";1;P$
  90 GO SUB 300
 100 FOR A=3 TO 19: PRINT AT A,1
;"    ";AT A,6;"
   ";AT A,26;"  ";AT A,29;"  "
 105 INPUT "CODIGO?:";C: IF C=99
99 THEN GO TO 200
 110 INPUT "NOMBRE?:";N$: IF LEN
N$>19 THEN GO TO 110
 120 INPUT "ES INGLESA?:";I$: IF
LEN I$>2 THEN GO TO 120
 130 INPUT "ORIGINAL O COPIA?:";
O$: IF LEN O$>2 THEN GO TO 130
 140 PRINT AT A,1;C;AT A,6;N$;AT
A,26;I$;AT A,29;O$
 150 INPUT "ES CORRECTO?(S/N):";
R$: IF R$="N" THEN GO TO 105
 160 PRINT #4;C'N$'I$'O$
 170 NEXT A: IF A>19 THEN LET A=
2: NEXT A
 190 STOP
 200 PRINT #4;C: CLOSE #4: CLS :
PRINT "FICHERO GRABADO": GO TO
20
 300 CLS : FOR a=0 TO 21: PRINT
AT a,0;"|";AT a,25;"|";AT a,5;"|
";AT a,28;"|";AT a,31;"|": NEXT
a
 310 LET A$="+----+-------------
------+--+--+"
 320 PRINT AT 0,0;A$;AT 2,0;A$;A
T 21,0;A$
 330 PRINT AT 1,2;"COD";AT 1,12;
"NOMBRE";AT 1,26;"IN";AT 1,29;"O
R": RETURN
 400 CLS : INPUT "NOMBRE DE EL F
ICHERO?:";Z$: OPEN #4;"M";1;Z$
 410 GO SUB 300
 420 FOR A=3 TO 19: PRINT AT A,1
;"    ";AT A,6;"
   ";AT A,26;"  ";AT A,29;"  "
 425 INPUT #4;C: IF C=9999 THEN
GO TO 480
 430 INPUT #4;N$;I$;O$
 440 PRINT AT A,1;C;AT A,6;N$;AT
A,26;I$;AT A,29;O$
 450 NEXT A: IF A>19 THEN LET A=
2: IF A=2 THEN GO SUB 1000: NEXT
A
 460 NEXT A
 470 STOP
 480 CLOSE #4: PAUSE 300: GO TO
20
 600 REM AQUI TENDRIA QUE CARGAR
SE EL C/M DE LA IMPRESORA SI UTI
LIZAS INTERFACE: PRINT : PRINT "
ESPERA"
 620 INPUT "NOMBRE DEL FICHERO?:
";F$: OPEN #4;"M";1;F$
 625 LPRINT " LISTADO DE PROGRAM
AS"
 630 INPUT #4;C
 640 IF C=9999 THEN GO TO 690
 650 INPUT #4;N$;I$;O$
 660 LPRINT C;"  ";N$
 670 GO TO 630
 690 CLOSE #4: GO TO 20
```

3

```
APER 6;AT f+2,g;"■■": LET f1=f:
LET g1=g
 130 LET ts=ts-2: IF ts=0 THEN L
ET tm=tm-1: LET ts=60: IF tm=-1
THEN GO TO 8500
 150 PRINT AT 17,27;tm;":";("0"
AND ts<10);ts
 400 IF IN 64510=254 THEN GO SUB
 1000
 410 IF IN 57342=254 THEN GO SUB
 1100
 450 IF g=x+1 THEN IF RND>sk THE
N GO SUB 2000
 500 LET y=y+(IN 61438=239 AND y
<17)-(IN 61438=247 AND y>2)
 510 LET x=x+(IN 61438=251 AND x
<22)-(IN 63486=239 AND x>7)
 550 LET f=f+INT (RND*2.5)-INT (
RND*2.5)+(f<y)-(f>y)
 553 IF f<2 THEN LET f=2
 555 IF f>17 THEN LET f=17
 560 LET g=g+INT (RND*2.5)-INT (
RND*2.5)+(g<x+2)-(g>x+2)
 563 IF g<x+1 THEN LET g=x+1
 565 IF g>23 THEN LET g=23
 570 IF NOT e THEN IF x>20 THEN
LET e=1
 580 IF e THEN LET e=e-(e-1 AND
x<20)+1: IF e=10 THEN GO SUB 300
0
 690 BEEP .01,-20
 700 GO TO 100
1000 PRINT INK 2;AT y,x+1;"■■":
BEEP .1,20
1010 PRINT INK 2;AT y,x+1;"■ "
1020 IF y=f+1 AND x=g-1 THEN LET
 s1=s1+1: BEEP .04,0: PRINT AT 8
,2;s1
1030 IF s1=k2 THEN GO TO 6000
1090 RETURN
1100 PRINT INK 2;AT y+2,x+1;"■■"
: BEEP .1,20
1110 PRINT INK 2;AT y+2,x+1;"■ "
1120 IF y=f-1 AND x=g-1 THEN LET
 s1=s1+1: BEEP .04,0: PRINT AT 8
,2;s1
1130 IF s1=k2 THEN GO TO 6000
1190 RETURN
2000 IF f<y THEN GO TO 2100
2010 PRINT AT f,g-1;"■■": BEEP .
1,30
2020 PRINT AT f,g-1;"■ "
2030 IF f=y+1 THEN LET s2=s2+1:
BEEP .04,0: PRINT AT 8,28;s2
2035 IF s2=k1 THEN GO TO 6500
2040 RETURN
2100 PRINT AT f+2,g-1;"■■": BEEP
 .1,30
2110 PRINT AT f+2,g-1;" ■"
2130 IF f=y-1 THEN LET s2=s2+1:
BEEP .04,0: PRINT AT 8,28;s2
2135 IF s2=k1 THEN GO TO 6500
2140 RETURN
3000 PRINT AT y1,x1;"  ";AT y1+1
,x1;"  ";AT y1+2,x1;"  "
3010 FOR x=19 TO 12 STEP -1: LET
 x1=x
3020 PRINT AT y1,x1; INK 2;"■■";
AT y1+2,x1;"■■";AT y1+1,x1; PAPE
R 0; INK 2;"■"; PAPER 6;" "; PAP
ER 2; INK 6;"■"; PAPER 6;"■";AT
y1,x1+2; INK 4;"■ ";AT y1+2,x1+2
;"■ "
3022 BEEP .02,0
3025 NEXT x
3030 PRINT AT y1,x1+2;" ";AT y1+
1,x1+2;" ";AT y1+2,x1+2;" "
3040 LET e=0: RETURN
6000 PRINT AT f,g;"■■";AT f+2,g
;"■■"; INK 2;AT f+1,g;"■"; INK
0;"■"; INK 6; PAPER 3;"■"
6010 PRINT FLASH 1;AT 1,7;"BIEN
BOXEADO. ESTA K.O.": BEEP .2,-20
```

(2)

```
 800 CLS : INPUT "NOMBRE DEL FIC
HERO ANTIGUO?:";M$: OPEN #4;"M";
1;M$
 810 INPUT "NOMBRE DE EL NUEVO F
ICHERO?:";K$: OPEN #5;"M";1;K$
 820 INPUT #4;C: IF C=9999 THEN
GO TO 860
 830 INPUT #4;N$;I$;O$
 840 PRINT #5;C'N$'I$'O$
 850 GO TO 820
 860 CLOSE #4: CLS : GO SUB 300
 870 FOR A=3 TO 19: PRINT AT A,1
;"    ";AT A,6;"    ";AT A,29;"  "
   ";AT A,26;"  "
 880 INPUT "CODIGO?:";C: IF C=99
99 THEN GO TO 970
 890 INPUT "NOMBRE?:";N$: IF LEN
N$>19 THEN GO TO 890
 900 INPUT "ES INGLESA?:";I$: IF
LEN I$>2 THEN GO TO 900
 910 INPUT "ORIGINAL O COPIA?:";
O$: IF LEN O$>2 THEN GO TO 910
 920 PRINT AT A,1;C;AT A,6;N$;AT
A,26;I$;AT A,29;O$
 930 INPUT "CORRECTO?:";R$: IF R
$="N" THEN GO TO 880
 940 PRINT #5;C'N$'I$'O$
 950 NEXT A: IF A>19 THEN LET A=
2: NEXT A
```

```
==================================
#Este es un programa para        #
#almacenar la lista de cintas    #
#que tiene, siga todas las       #
#instrucciones como se indica    #
#                                #
#RECUERDE:                       #
#-Tener siempre cartucho puesto  #
#-Trabajar en mayusculas.        #
#-Al anadir fichas desaparece    #
# el fichero antiguo.            #
#-Por la impresora no aparecen   #
# or/cop ni si es inglesa.       #
#-En caso de "BREAK debe cerrar  #
# los ficheros (CLOSE# 4)        #
#                                #
#                                #
#PULSA TECLA                     #
#                                #
==================================
```

```
 960 STOP
 970 PRINT #5;C: CLOSE #5: CLS :
PRINT "NUEVO FICHERO GRABADO"
```

```
6020 GO TO 6600
6500 PRINT AT y,x-1; INK 2;"■■"
;AT y+2,x-1;"■■"; INK 6; PAPER
2;AT y+1,x-2;" "; PAPER 6; INK 2
;"■"; INK 4;"■"
6510 PRINT AT 1,7; FLASH 1;"MALA
 SUERTE- K.O.!"
6600 FOR f=1 TO 200: NEXT f
6620 CLS : IF k1=s2 OR s2>s1 THE
N GO TO 6700
6630 PRINT "Bien hecho ! Acabast
e con ";b$;" en ";round;" rondas
."
6640 PRINT INK 3'"El resultado f
ue de ";s1;" puntos","por tu par
te y ";s2;" por ";b$
6650 PRINT INK 2'"Deberias de es
coger un rival masfuerte."
6660 GO TO 6780
6700 PRINT "Mala suerte ! Te ani
quilaron en ";round," rondas."
6710 PRINT INK 3'"El resultado f
ue de ";s1;" puntos","por tu par
te y ";s2;" por ";b$
6740 PRINT INK 2'"Deberias de es
coger un rival masdebil."
6780 PRINT "Elige uno de estos o
ponentes:": RESTORE 7500: PRINT
: FOR f=1 TO 10: READ a$: PRINT
f;"...";("." AND f<>10);a$: NEXT
 f
6790 INPUT INK 7; PAPER 0;"Intro
duce el numero de tu rival (o 0
si ya estas harto). ";x
6800 IF x=0 THEN STOP
6810 CLS : RESTORE 7500: FOR f=1
 TO x: READ a$: NEXT f: LET b$=a
$(1 TO 6): PRINT "OK - Lucharas
con ";b$: PRINT INVERSE 1' INK 3
'"PULSA CUALQUIER TECLA PARA
   COMENZAR EL COMBATE."
6820 PAUSE 0: CLS
6830 LET s1=0: LET s2=s1: LET sk
=(11-x)/12: LET k2=10+x+INT (RND
*6): LET k1=16-x+INT (RND*10): G
O TO 15
7000 RESTORE 7500
```

```
7100 PRINT AT 10,0;"Escribe tu n
ombre (longitud: 6  letras max.)
": INPUT c$: IF LEN c$>6 THEN GO
 TO 7100
7200 CLS
7500 DATA "Henry Hopeless","Arno
ld Awful","Nigel No-Good","Phili
p Fair","Andrew Average","Roger
Right-Hook","Brian Brick-Wall","
Simon Super","Edward Excellent",
"Flynn Fantastic"
7510 PRINT INVERSE 1;"      C A M
 P E O N A T O             DE B
 O X E O              "; INVERSE
 0;"            @ PAUL STANLEY"
7520 PRINT INK 3'"Verdad que qui
eres llegar a ser un campeon ? P
ues este es el     juego que te h
ace falta."; INK 2'"Con que opon
ente quieres enfren-tarte primer
o ? "
7530 PRINT : FOR f=1 TO 10: READ
 a$: PRINT INK 1;f;"...";("." AN
D f<>10);a$: NEXT f
7540 INPUT x: IF x<1 OR x>10 THE
N GO TO 7540
7545 LET sk=(11-x)/12
7546 LET k2=10+x+INT (RND*6): LE
T k1=16-x+INT (RND*10)
7550 RESTORE 7500: FOR f=1 TO x:
 READ a$: NEXT f: LET b$=a$(1 TO
 6)
7560 CLS : PRINT "Muy bien, tu r
ival sera ",a$' INK 2'"Cada comb
ate constara de solo    3 rondas.
 Para moverte por el    ring usa
5,6,7 y 8 y golpea con Q(mano iz
quierda) y P(mano       derecha).
 Ten en cuenta que      puedes mo
verte y golpear a la    vez."
7570 PRINT ' INK 1;"Solo puntuan
 los golpes a la    cabeza. Pued
en llegar a dejar   K.O."
7580 PRINT ' INVERSE 1;"PULSA CU
ALQUIER TECLA PARA      COMENZAR
 EL COMBATE.": PAUSE 0: CLS
7590 RETURN
```

```
================================
#                              #
#                              #
#  A CREAR FICHERO             #
#                              #
#  B LISTAR FICHERO            #
#                              #
#  C IMPRIMIR FICHERO (con rs) #
#                              #
#  D ANADIR FICHAS             #
#                              #
#                              #
================================
```

```
+---+------------------+--+--+
|COD|      NOMBRE      |IN|OR|
+---+------------------+--+--+
|1  |THE HOBBIT        |SI|OR|
|2  |AJEDREZ           |NO|CO|
|5  |MARCIANOS         |SI|OR|
|   |                  |  |  |
+---+------------------+--+--+
```

```
980 ERASE "M";1;M$: PRINT "FICH
ERO ANTIGUO BORRADO": PAUSE 150:
 GO TO 20
```

```
1000 LET J$=INKEY$: IF J$="" THE
N GO TO 1000
1010 RETURN
```

3

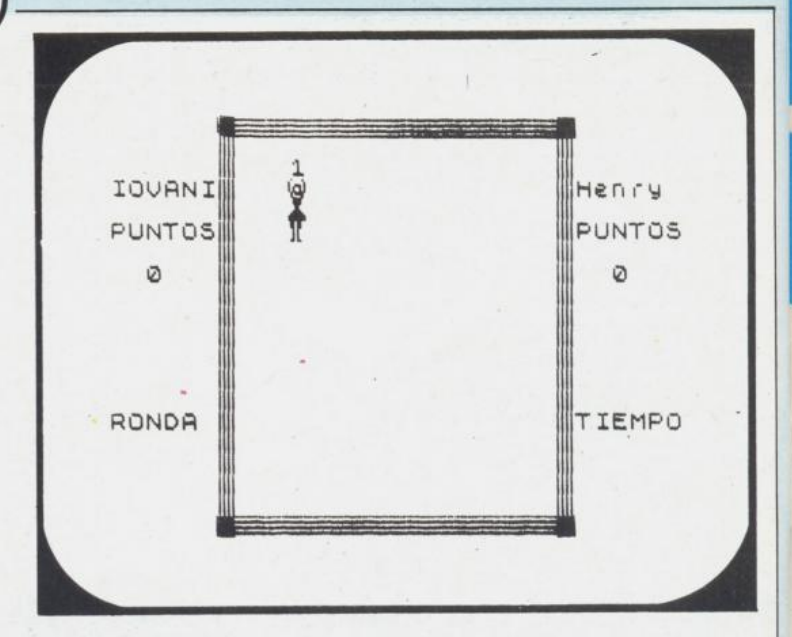
```
8500 PRINT AT 17,27;"0:00"
8505 BEEP .25,0: BEEP .25,0
8510 LET round=round+1: IF round
=4 THEN LET round=3: GO TO 6620
8520 PRINT AT 21,0;"Preparate pa
ra la ronda ";round;"."; INK 6;
PAPER 0;AT 2,7;"::";AT 19,24;"::":
 FOR i=1 TO 20
8525 LET y=y-(y>2): LET x=x-(x>7
)
8530 PRINT AT y1,x1;"  ";AT y1+1
,x1;"  ";AT y1+2,x1;"  "; INK 2;A
T y,x;"■"; PAPER 0;AT y+1,x;"::"
; PAPER 6;AT y+2,x;"■": LET y1=
y: LET x1=x
```



```
8535 LET f=f+(f<17): LET g=g+(g<
23)
8540 PRINT AT f1,g1;"  ";AT f1+1
,g1+1;"  ";AT f1+2,g1;"  ";AT f,g
;"■"; PAPER 7;AT f+1,g+1;"::"; P
APER 6;AT f+2,g;"■": LET f1=f:
LET g1=g
8542 IF ATTR (2,7)=50 THEN IF AT
TR (19,24)=48 THEN GO SUB 9550:
FOR f=-1 TO 2: PRINT AT y1+f,x1;
"  ": NEXT f: PRINT AT 21,0;"
                              ": GO
 TO 62
8545 BEEP .1,-18: NEXT i
9000 RESTORE 9010: LET f=0: LET
g=255: FOR y=USR "a" TO USR "t"+
7: READ x: POKE y,x: NEXT y
9010 DATA f,63,127,g,g,252,254,2
54,f,g,g,g,g,f,f,f,28,254,254,25
4,254,28,f,f
9020 DATA 195,129,f,f,f,f,129,19
5,254,254,252,g,g,127,63,f,f,f,f
,g,g,g,g,f
9030 DATA f,f,28,254,254,254,254
,28,56,127,127,127,127,56,f,f
9035 DATA f,252,254,g,g,63,127,1
27,f,f,56,127,127,127,127,56,127
,127,63,g,g,254,252,f
9045 DATA 66,129,157,165,165,94,
66,60,60,60,24,24,60,126,g,g,108
,108,108,40,40,40,40,108
9050 DATA 28,30,15,15,31,62,126,
252,248,240,240,224,224,240,240,
248,252,126,62,31,15,15,30,28
9060 DATA 56,120,240,240,248,124
,126,63,31,15,15,7,7,15,15,31,63
,126,124,248,240,240,120,56
9100 RETURN
9500 INK 4: FOR f=0 TO 7 STEP 2:
 PLOT 55-f,15-f: DRAW 0,145+2*f:
 DRAW 145+2*f,0: DRAW 0,-2*f-145
: DRAW -2*f-145,0: NEXT f
9510 PRINT INK 3; BRIGHT 1;AT 1,
6;"■";AT 1,25;"■";AT 20,6;"■";AT
 20,25;"■"
```

③

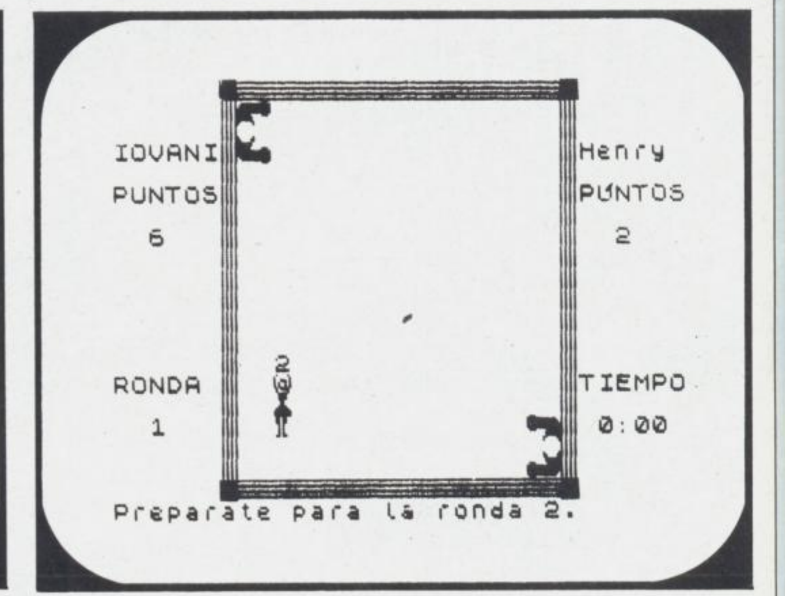
```
9520 INK 0
9525 PRINT AT 4,0;c$;AT 4,26;b$
9530 PRINT AT 6,0; INK 2;"PUNTOS
"; INK 0;AT 6,26;"PUNTOS"; INK 3
;AT 15,0;"RONDA";AT 15,26;"TIEMP
O"
9540 PRINT AT 8,2;s1;AT 8,28;s2
9545 RETURN
9550 LET y1=16: LET x1=9: RESTOR
E 9560: FOR f=1 TO 4: READ y,x
9560 DATA 0,1,-1,0,0,-1,1,0
9570 FOR e=1 TO 12: PRINT AT y1,
x1;"@";AT y1+1,x1;"♣";AT y1+2,x1
;"Ⅱ"; PAPER 2; INK 7;AT y1-1,x1;
round
9580 BEEP .06,-45
9585 IF e=12 THEN FOR o=25 TO 35
: BEEP .005,o: NEXT o: FOR o=1 T
O 20: NEXT o: FOR o=35 TO 25 STE
P -1: BEEP .005,o: NEXT o
9590 PRINT AT y1,x1;" ";AT y1+1,
x1;" ";AT y1+2,x1;" ";AT y1-1,x1
;"  "
9600 LET y1=y1+y: LET x1=x1+x: N
EXT e: NEXT f
9610 RETURN
```




```
        C A M P E O N A T O
        DE  B O X E O
            © PAUL STANLEY

Verdad que quieres llegar a ser
un campeon ? Pues este es el
juego que te hace falta.
Con que oponente quieres enfren-
tarte primero ?

1....Henry Hopeless
2....Arnold Awful
3....Nigel No-Good
4....Philip Fair
5....Andrew Average
6....Roger Right-Hook
7....Brian Brick-Wall
8....Simon Super
9....Edward Excellent
10...Flynn Fantastic
```

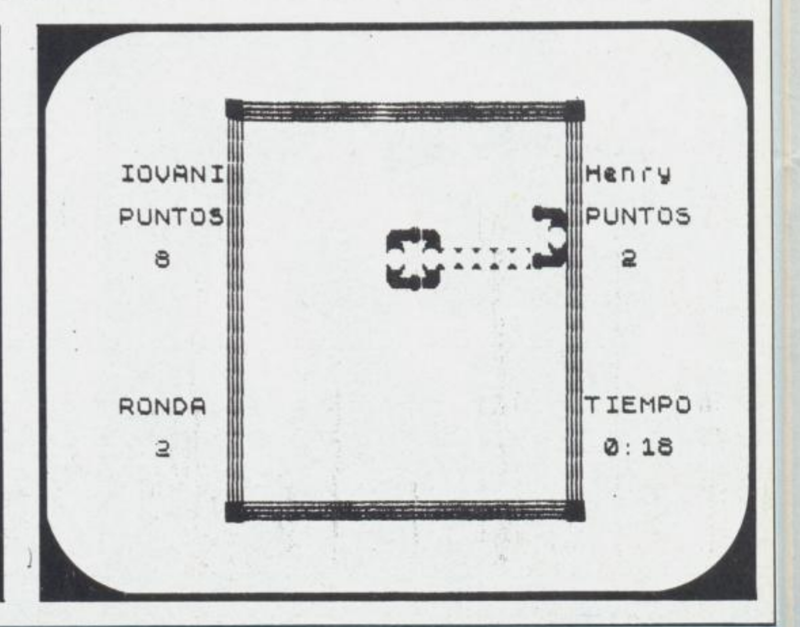



```
Muy bien, tu rival sera
Henry Hopeless

Cada combate constara de solo
3 rondas. Para moverte por el
ring usa 5,6,7 y 8 y golpea con
Q(mano izquierda) y P(mano
derecha). Ten en cuenta que
puedes moverte y golpear a la
vez.

Solo puntuan los golpes a la
cabeza. Pueden llegar a dejar
K.O.

PULSA CUALQUIER TECLA PARA
COMENZAR EL COMBATE.
```



Minas

¿Qué le parece ir a recoger huevos de oro? Ese es el objetivo de este juego, recoger el máximo posible, pero no se lo pondrá nada fácil, pues su camino se encuentra plagado de minas e incluso con un pequeño enemigo que corre a su encuentro con muy malas intenciones. Afortunadamente en su camino podrá recoger hachas con las que destruir al enemigo, explosionar una bomba o destruir un muro.

Aunque parezca sencillo le aseguramos que no es fácil hacer una buena colecta de huevos: dispone de pocos hachas y un número de vidas limitado.

Notas gráficas

ABCDEFGHIJKLMNOPQRSTU

◇☓☓✕●✱/HIJKLMNOPQRSTU

Autor: P. Stanley 16K

```
   1 RANDOMIZE : BORDER 0: PAPER
 0: INK 7: CLS
   5 REM © P. Stanley (21.3.83)
  10 GO SUB 9000
  15 LET hs=0
  20 GO TO 6030
  50 BEEP .1,-10
  55 LET li=3
  60 PRINT AT 0,13;"☓☓   Max.Pun
t.";hs
 100 PRINT AT y1,x1;" "; INK 6;A
T y,x;a$: LET y1=y: LET x1=x
 110 LET x=x+(IN 57342=254)-(IN
57342=251): LET y=y+(IN 65278=25
3)-(IN 64510=254)
 120 LET a=ATTR (y,x): IF SCREEN
$ (y,x)=" " THEN GO TO 700
 122 IF a=7 THEN IF a$="☓" THEN
GO TO 5000
 125 IF a=7 THEN LET y=y1: LET x
=x1
 130 IF a=70 THEN BEEP .01,20: L
ET sc=sc+10: PRINT INK 7;AT 0,6;
sc: IF sc/500=INT (sc/500) THEN
PRINT AT 0,12+li; FLASH 1;"☓": F
OR g=1 TO 5: BEEP .2,-30: NEXT g
: PRINT AT 0,12+li;"☓": LET li=l
i+1
 140 IF a=5 THEN GO TO 6000
 150 IF a=68 THEN BEEP .1,10: LE
T a$="☓"
 700 PRINT AT my,mx;" ": IF RND>
sk THEN PRINT AT my1,mx1; INK 5;
"✱"
 705 LET my1=my: LET mx1=mx
 710 LET my=my+(.5 AND my<y)-(.5
 AND my>y): LET mx=mx+(.5 AND mx
<x)-(.5 AND mx>x)
 715 IF ATTR (my,mx)=5 THEN GO T
O 6000
 720 PRINT AT my,mx; INK 5; BRIG
HT 1;"✕"
 940 IF RND<.2 THEN PRINT AT RND
*18+2,RND*28+1; INK 6; BRIGHT 1;
"●": IF RND>.9 THEN PRINT INK 4;
BRIGHT 1;AT RND*18+2,RND*28+1;"
/"
 999 GO TO 100
5000 IF y=1 OR y=21 OR x=0 OR x=
30 THEN GO TO 125
5005 FOR f=50 TO 0 STEP -5: BEEP
 .02,f: NEXT f
5010 LET a$="☓"
5030 GO TO 130
5500 PRINT AT 0,12+li;" ": PRINT
 AT y,x;" ": LET y=2: LET x=1
5600 LET my=INT (RND*11)+10: LET
 mx=INT (RND*18)+12: GO TO 100
6000 PRINT AT my,mx;" ": LET my=
INT (RND*11)+10: LET mx=(RND*18)
+12: LET my1=my: LET mx1=mx: IF
a$="☓" THEN LET a$="☓": FOR f=0
TO 40 STEP 5: BEEP .04,f: NEXT f
: GO TO 700
6005 PRINT AT y1,x1;" ": FOR f=1
 TO 7: FOR g=7 TO 0 STEP -1: PRI
NT INK g;AT y,x;"☓": BEEP .02,g*
f: NEXT g: NEXT f
6007 LET li=li-1: IF li<>0 THEN
GO TO 5500
6010 FOR g=0 TO 60: NEXT g
6020 CLS : PRINT AT 5,8; INVERSE
 1;"SE ACABO!!!"
6025 PRINT AT 8,9; INK 5; BRIGHT
 1;"Puntos.... ";sc
6026 IF sc>hs THEN LET hs=sc: PR
INT ''" Bien hecho- Otro juego!!
": GO TO 6030
6027 PRINT ''"     Maxima puntuac
ion ";hs
6030 PRINT INK 6''''"  Presiona
1 (dificil) o 2 (fa-   cil)"
6040 IF INKEY$<>"1" AND INKEY$<>
"2" THEN GO TO 6040
6045 LET sk=(.7 AND INKEY$="1")+
(.85 AND INKEY$="2")
6050 CLS
7000 FOR f=1 TO 21: PRINT AT f,0
;"◇";AT f,30;"◇": NEXT f
7010 FOR f=1 TO 29: PRINT AT 1,f
;"◇";AT 21,f;"◇": NEXT f
7020 FOR g=1 TO (7 AND sk=.85)+(
14 AND sk=.7): LET s=RND*28: LET
 f=s+RND*10: LET x=INT (RND*9)*2
+1: IF f>29 THEN LET f=29
7030 FOR y=s TO f: PRINT AT x,y;
"◇"

7040 NEXT y
7050 LET s=RND*17+2: LET f=s+RND
*10: LET x=INT (RND*14)*2+2: IF
```



(4)

```
f>20 THEN LET f=20
7050 FOR y=s TO f: PRINT AT y,x;
"◇": NEXT y: NEXT g
7390 FOR f=1 TO 20
7400 LET y=RND*18+2: LET x=RND*2
8+1
7410 IF SCREEN$ (y,x)="" THEN GO
 TO 7400
7420 PRINT BRIGHT 1; INK 6;AT y,
x;"•": NEXT f
7500 FOR f=1 TO 10
7510 LET y=RND*18+2: LET x=RND*2
8+1
7520 IF SCREEN$ (y,x)="" THEN GO
 TO 7510
7540 PRINT INK 5;AT y,x;"*": NEX
T f
7550 FOR f=1 TO 5: PRINT AT RND*
18+2,RND*28+1; INK 4; BRIGHT 1;"
/": NEXT f
7800 LET y=2: LET x=1: LET y1=y:
 LET x1=x: LET a$="*"
7840 LET my=INT (RND*11)+10: LET
 mx=INT (RND*18)+12: LET my1=my:
 LET mx1=mx
7850 LET sc=0: PRINT AT 0,0;"Pun
tos ";sc
7999 GO TO 50
9000 FOR f=USR "a" TO USR "g"+7:
 READ g: POKE f,g: NEXT f
9010 DATA 24,36,66,153,153,66,36
,24,28,28,73,62,8,28,20,54
9020 DATA 220,220,73,126,72,92,2
0,54,195,36,24,36,126,90,129,0
9030 DATA 0,0,0,12,62,126,62,12,
0,0,8,73,42,28,62,127
9040 DATA 4,14,6,12,24,48,96,0
9500 PRINT INK 0;AT 0,0;" _ _ __
_ _ _ ___  _ ___ _ _  _ _ _
  _ _ _    _ _ _ _ _ _ _ _ _
  _ _ ___  _ ___ _ _ _ _ _ _
  _ _ _    _ _ _  _ "_ _ _ __
9600 FOR f=0 TO 4: PRINT INK 5;A
T f,0; BRIGHT 1;"X": FOR g=1 TO
31
9610 PRINT AT f,g-1; OVER 1; INK
 0; BRIGHT 1;"X"; INK 5;"X"
9620 IF SCREEN$ (f,g-1)="_" THEN
 PRINT INK 2; PAPER 7;AT f,g-1;"
*"
9630 BEEP .04,(g+f)/3
9640 NEXT g: PRINT AT f,31;" ":
NEXT f
9645 PLOT 16,173: DRAW 3,-10: DR
AW 4,10: PLOT 80,173: DRAW 7,-28
9650 PRINT '"Recorre el laberint
o recogiendo huevos, evitando la
s minas y al enemigo."
9700 PRINT INK 6'"Si coges un ha
cha podras des-   truir los muro
s, minas e inclu- so al enemigo
que avanza."
9750 PRINT '"Dispon💧es de 3 vidas
, pero se te dara una mas cada 5
```

64 Caracteres

Aunque no sea usted un fanático de la lingüística, más de una vez se habrá preguntado cómo podría hacer que su ordenador pusiese acentos o tuviese caracteres españoles como la Ñ que evitasen el embarazoso problema de poner ano en vez de año, por ejemplo. La cosa es sencilla: ¡sólo ha de crearse su propio alfabeto! Y puestos en materia ¿por qué no cambiar a un tipo de escritura que permita 64 caracteres por línea?

Todo es posible con este programa. Tiene todos los caracteres del código ASCII excepto los que se acceden en modo extendido: la ñ se obtiene con SYMBOL *SHIFT* y 2, la Ñ con SYMBOL *SHIFT* y 6, las vocales acentuadas se obtienen pulsando primero *CAPS SHIFT* +SYMBOL *SHIFT*, y después la vocal correspondiente: se puede borrar con *DELETE*; tiene un *BEEP* de aviso de fin de línea, otro de última línea y un tercero de pantalla llena.

El texto escrito en la pantalla se podrá pasar a impresora o almacenar en cinta de cassette, si se elije esta segunda opción se puede guardar en una cinta de sesenta minutos aproximadamente 90 pantallas por cada cara, lo que equivale a un total de 1.932 líneas de 64 caracteres, es decir, el contenido de un libro de bolsillo. El programa permite también leer estas cintas para lo cual pregunta al principio si se desea leer o escribir.

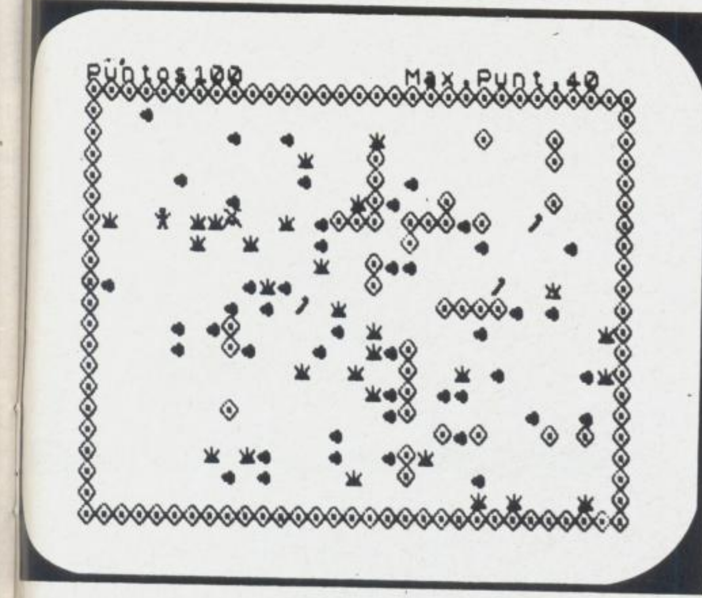
Este pequeño editor utiliza por tanto un juego de caracteres propio que se almacena por duplicado y que habrá de ser introducido manualmente con ayuda del programa "cargador de caracteres" que se adjunta, de acuerdo con la tabla que se incluye a continuación.

Una vez más queda demostrado que los límites en materia informática sólo están en su imaginación.

```
   10 REM Carga Caracteres
   20 CLEAR 29663
   30 FOR f=31832 TO 32592 STEP 8
   40 CLS : PRINT "DIRECCION: ";f
   50 FOR n=0 TO 7
   60 INPUT a
   70 POKE f+n,a: POKE f+n-768,a*
16: PRINT a;" ";
   80 NEXT n: BEEP .01,0
   90 NEXT f
  100 CLS : PRINT "O.K. Caractere
s cargados": STOP
```

Autor: Jesús Alonso Rodríguez — 16K

```
    4 REM 64 Caracteres por linea
    6 REM  © Jesus Alonso 1.984
    8 REM  Carga Caracteres Cinta
   10 PAPER 0: BORDER 1: INK 7: C
LS
   20 CLEAR 29663
   30 PRINT AT 10,6;"CARGANDO CAR
ACTERES"
   35 INK 0
   40 LOAD "caracteres"CODE 31064
,1536
   45 INK 7
   50 PRINT AT 10,6;"PARE LA CINT
A     ";AT 12,0;"     PULSE CUA
LQUIER TECLA": PAUSE 0
  100 REM Inicializacion
  110 INPUT "LEER O ESCRIBIR ? ";
b$
  120 IF b$(1)="l" THEN GO TO 400
```



MINE LAY

Recorre el laberinto recogiendo huevos, evitando las minas y al enemigo.

Si coges un hacha podras destruir los muros, minas e incluso al enemigo que avanza.

Dispones de 3 vidas, pero se te dara una mas cada 500 puntos.

Q.....Arriba Z.....Abajo

I.....Izquierda P....Derecha

```
00 puntos."
9800 PRINT INK 6'" Q.....Arriba
     Z.....Abajo"' INK 5'" I....
.Izquierda    P....Derecha"
9900 INPUT "Presiona ENTER para
comenzar....";a$
9910 IF INKEY$="" THEN GO TO 991
0
9920 CLS
9999 RETURN
```

```
0
 130 IF b$(1)="e" THEN GO TO 150
 140 GO TO 110
 150 CLS : PRINT AT 0,0;"█";CHR$
8;: BEEP .01,0
 160 LET m=29664: LET c=0
 900 REM Bucle escribir
1000 PAUSE 0: LET a=CODE INKEY$:
 BEEP .01,0
1005 IF a<12 THEN GO TO 1000
1010 IF a=13 THEN PRINT " "'"█";
CHR$ 8;: POKE m,13: LET m=m+1: G
O TO 1190
1020 IF a=12 THEN GO TO 2500
1030 IF a=14 THEN LET c=1: GO TO
 1000
1040 IF a=77 OR a=87 OR a=109 OR
 a=119 THEN GO TO 2300
1045 IF c=1 THEN LET a=123*(a=97
)+124*(a=101)+125*(a=105)+126*(a
=111)+127*(a=117): LET c=0: IF a
=0 THEN GO TO 1000
1050 POKE m,a: LET m=m+1: GO SUB
 2000
1100 PAUSE 0: LET a=CODE INKEY$:
 BEEP .01,0
1105 IF a<12 THEN GO TO 1100
1110 IF a=13 THEN PRINT CHR$ b'"
█";CHR$ 8;: POKE m,13: LET m=m+1
: GO TO 1190
1120 IF a=12 THEN GO TO 2600
1130 IF a=14 THEN LET c=1: GO TO
 1100
1140 IF a=77 OR a=87 OR a=109 OR
 a=119 THEN GO TO 2400
1145 IF c=1 THEN LET a=123*(a=97
)+124*(a=101)+125*(a=105)+126*(a
=111)+127*(a=117): LET c=0: IF a
=0 THEN GO TO 1100
1150 POKE m,a: LET m=m+1: GO SUB
 2100
1160 IF PEEK 23688=7 THEN BEEP .
1,40
1190 IF PEEK 23689=4 AND PEEK 23
688=33 THEN BEEP .3,20
1200 IF PEEK 23689=3 THEN POKE m
,255: BEEP .6,20: GO TO 3000
1210 GO TO 1000
2000 REM Sub Print - A
2010 POKE 23606,88: POKE 23607,1
20: LET b=a
2020 PRINT CHR$ a;CHR$ 8; OVER 1
;"█";CHR$ 8;
2030 RETURN
2100 REM Sub. Print - B
2110 PRINT CHR$ b;CHR$ 8;
2120 POKE 23606,88: POKE 23607,1
23
2130 PRINT OVER 1;CHR$ a;"█";CHR
$ 8;
2140 RETURN
2300 REM Sub. m,M,w,W - A
2310 POKE m,a: LET m=m+1: GO SUB
 2000
2320 LET a=49*(a=77)+91*(a=87)+9
2*(a=109)+93*(a=119)
2330 POKE m,a: LET m=m+1: GO SUB
 2100
2340 GO TO 1160
2400 REM Sub. m,M,w,W - B
2410 POKE m,a: LET m=m+1: GO SUB
 2100
2420 LET a=49*(a=77)+91*(a=87)+9
2*(a=109)+93*(a=119)
2430 POKE m,a: LET m=m+1: GO SUB
 2000
2440 GO TO 1100
2500 REM Sub. DELETE - A
2505 POKE m,12: LET m=m+1
2510 IF PEEK 23689=24 AND PEEK 2
3688=33 THEN GO TO 1000
2520 PRINT CHR$ 8;"█ ";CHR$ 8;CH
R$ 8;
2530 GO TO 1000
2600 REM Sub. DELETE - B
2605 POKE m,12: LET m=m+1
2610 PRINT "█";CHR$ 8;
```

(5)

```
2620 GO TO 1000
3000 REM Sub. Pantalla llena
3005 POKE 23606,0: POKE 23607,60
3010 PRINT " "
3020 INPUT "DESEA IMPRIMIRLO ? "
;b$
3030 IF b$(1)="s" THEN COPY
3040 INPUT "DESEA GUARDARLO EN C
INTA ? ";b$
3050 IF b$(1)="s" THEN GO SUB 31
00
3060 INPUT "DESEA BORRARLO ? ";b
$
3070 IF b$(1)="s" THEN GO TO 100
3080 GO TO 3020
3100 REM Sub. Texto a cinta
3110 SAVE "texto"CODE 29664,1400
3120 VERIFY "texto"CODE 29664,14
00
3130 RETURN
4000 REM Bucle lectura
```

```
4010 INPUT "PREPARE LA CINTA Y P
ULSE 'ENTER'";b$
4020 LOAD "texto"CODE 29664,1400
4030 INPUT "PARE LA CINTA Y PULS
E 'ENTER'";b$
4040 CLS : PRINT AT 0,0;"▌";CHR$
 8;
4050 LET m=29664
4100 LET a=PEEK m: LET m=m+1
4105 IF a<12 THEN GO TO 4100
4110 IF a=255 THEN GO TO 4500
4120 IF a=13 THEN PRINT " "'"▌";
CHR$ 8;: GO TO 4100
4130 IF a=12 THEN PRINT CHR$ 8;"
▌ ";CHR$ 8;CHR$ 8;: GO TO 4100
4140 GO SUB 2000
4200 LET a=PEEK m: LET m=m+1
4205 IF a<12 THEN GO TO 4200
4210 IF a=255 THEN GO TO 4500
4220 IF a=13 THEN PRINT CHR$ b'"
▌";CHR$ 8:: GO TO 4100
```

JUEGO DE CARACTERES

DIRECC.	DATOS							
31832 —	0	0	0	0	0	0	0	0
31840 —	0	4	4	4	4	0	4	0
31848 —	0	10	10	0	0	0	0	0
31856 —	0	0	4	14	4	14	4	0
31864 —	4	14	8	12	6	2	14	4
31872 —	0	0	9	2	4	9	0	0
31880 —	6	0	9	13	11	9	9	0
31888 —	2	4	0	0	0	0	0	0
31896 —	2	4	8	8	8	8	4	2
31904 —	8	4	2	2	2	2	4	8
31912 —	0	0	0	10	4	10	0	0
31920 —	0	0	0	4	14	4	0	0
31928 —	0	0	0	0	2	2	4	0
31936 —	0	0	0	0	14	0	0	0
31944 —	0	0	0	0	0	0	2	0
31952 —	0	0	1	2	4	8	0	0
31960 —	0	14	10	10	10	10	14	0
31968 —	0	2	6	10	2	2	2	0
31976 —	0	14	2	2	14	8	14	0
31984 —	0	14	2	2	14	8	14	0
31992 —	0	10	10	14	2	2	2	0
32000 —	0	14	8	14	2	2	14	0
32008 —	0	12	8	8	14	10	14	0
32016 —	0	14	2	2	2	2	2	0
32024 —	0	14	10	10	14	10	14	0
32032 —	0	14	10	10	14	2	2	0
32040 —	0	0	4	0	0	4	0	0
32048 —	0	2	0	0	2	2	4	0
32056 —	0	1	2	4	8	4	2	1
32064 —	0	0	0	14	0	14	0	0
32072 —	0	8	4	2	1	2	4	8
32080 —	0	14	10	2	4	0	4	0
32088 —	0	6	0	12	10	10	10	0
32096 —	0	14	10	10	14	10	10	0
32104 —	0	12	10	12	10	10	12	0
32112 —	0	14	10	8	8	10	14	0
32120 —	0	12	10	10	10	10	12	0
32128 —	0	14	8	12	8	8	14	0
32136 —	0	14	8	12	8	8	8	0
32144 —	0	14	8	8	10	10	14	0
32152 —	0	10	10	14	14	10	10	0
32160 —	0	4	4	4	4	4	4	0
32168 —	0	2	2	2	2	10	14	0
32176 —	0	10	12	12	10	10	10	0
32184 —	0	8	8	8	8	8	14	0
32192 —	0	4	6	5	4	4	4	0
32200 —	0	9	13	11	9	9	9	0
32208 —	0	4	10	10	10	10	4	0
32216 —	0	14	10	14	8	8	8	0
32224 —	0	4	10	10	10	14	6	1
32232 —	0	12	10	10	12	10	9	0
32240 —	0	14	10	8	4	2	14	0
32248 —	0	14	4	4	4	4	4	0
32256 —	0	10	10	10	10	10	14	0
32264 —	0	10	10	10	10	10	4	0
32272 —	0	4	4	4	4	5	2	0
32280 —	0	10	10	4	4	10	10	0
32288 —	0	10	10	10	4	4	4	0
32296 —	0	14	2	4	4	8	14	0
32304 —	0	2	2	2	2	10	4	0
32312 —	0	0	8	4	4	4	4	0
32320 —	0	0	4	4	4	4	8	0
32328 —	0	4	10	0	4	4	4	0
32336 —	0	0	0	0	0	0	15	0
32344 —	0	4	10	8	12	8	14	0
32352 —	0	0	14	2	14	10	15	0
32360 —	8	8	14	10	10	10	14	0
32368 —	0	0	14	8	8	8	14	0
32376 —	2	2	14	10	10	10	14	0
32384 —	0	0	14	10	14	8	14	0
32392 —	0	0	6	4	14	4	4	4
32400 —	0	0	14	10	10	14	2	14
32408 —	8	8	14	10	10	10	10	0
32416 —	0	4	0	4	4	4	6	0
32424 —	0	2	0	2	2	2	2	14
32432 —	8	8	10	12	12	10	10	0
32440 —	12	4	4	4	4	4	14	0
32448 —	0	0	6	5	5	5	5	0
32456 —	0	0	12	10	10	10	10	0
32464 —	0	0	14	10	10	10	14	0
32472 —	0	0	14	10	10	14	8	8
32480 —	0	2	14	10	10	14	2	2
32488 —	0	0	6	8	8	8	8	0
32496 —	0	0	14	8	4	2	14	0
32504 —	4	4	14	4	4	4	6	0
32512 —	0	0	10	10	10	10	15	0
32520 —	0	0	10	10	10	10	4	0
32528 —	0	0	4	5	5	5	2	0
32536 —	0	0	10	10	4	4	10	0
32544 —	0	0	10	10	10	14	2	14
32552 —	0	0	14	2	4	8	14	0
32560 —	2	4	14	2	14	10	15	0
32568 —	2	4	14	10	14	8	14	0
32576 —	2	4	0	4	4	4	6	0
32584 —	2	4	14	10	10	10	14	0
32592 —	2	4	10	10	10	10	15	0

```
4230 IF a=12 THEN PRINT "▌";CHR$
 8;: GO TO 4100
4240 GO SUB 2100
4250 IF m>=31063 THEN GO TO 4500
4260 GO TO 4100
4500 POKE 23606,0: POKE 23607,60
4510 INPUT "DESEA LEER OTRO TEXT
O ? ";b$
4520 IF b$(1)="s" THEN GO TO 400
0
4530 GO TO 100
9000 REM Salva en cinta
9010 SAVE "escribir" LINE 0
9020 SAVE "caracteres"CODE 31064
,1536
9030 VERIFY "escribir"
9040 VERIFY "caracteres"CODE 310
64,1536
```

Estaba sentado en la puerta de su casa, cuando aparecierón aquellos extraterrestres mostrandose agresiva y covardemente.

Gentes del lugar aseguran igualmente haber sido testigos de importantes avistamientos.

Cáceres, 25 de Mayo de 1984

Resumen de actividades del año

	Primer semestre	Segundo semestre
Ventas	250.000	400.000
Costes	150.000	270.00
Margen	100.000	130.00

Test de inglés

Aprender inglés ha dejado de ser una cuestión de moda para convertirse en una auténtica necesidad. Para aprenderlo puede optar por muchos diferentes caminos, pero el que le ofrecemos con este programa es más divertido. Como no se lo vamos a vender, le podemos decir que con él no aprenderá gran cosa, pero seguro que pasa un rato agradable e instructivo.

El programa le formulará preguntas sobre los verbos irregulares que previamente habrá de introducir (se almacenan en una variable alfanumérica de tres dimensiones a fin de guardar los tres tiempos de este tipo de verbos: infinitivo, pasado y participio). Se puede seleccionar el tipo de pregunta (siempre el mismo tiempo, generalmente el infinitivo, o al azar).

Autor: Alfonso Martín — 16K




```
   1 REM verbos irregulares en
ingles
   5 BORDER 7: PAPER 6: INK 0
   7 INPUT " tienes los verbos i
ntroducidos?"; INK 2;s$
   8 IF s$="s" THEN GO TO 200
  10 PRINT ,,"vamos a introducir
los segun te  los vaya pidiendo.
 primero dime cuantos hay"
  15 INPUT n
  20 DIM a$(n,3,10)
  30 FOR k=1 TO n
  35 FOR l=1 TO 3
  40 PRINT ,,"introduce el tiemp
o ";("infinitivo" AND l=1);("pas
ado" AND l=2);("participio" AND
l=3)
  50 INPUT a$(k,l)
  55 PRINT ,a$(k,l)
  60 NEXT l
  61 PRINT AT 19,0;"si hay algun
 error pulsa "; INK 1;"E"; INK 2
  62 FOR j=1 TO 400
  63 IF INKEY$="e" THEN GO SUB 1
00
  64 NEXT j
  65 CLS
  67 PRINT ,,"ahora otro"
  70 NEXT k
  80 GO TO 200
 100 CLS
 110 INPUT "donde esta el error:
inf(1), pa-sado(2), part.(3)";h
 130 PRINT ,,"introduce el tiemp
o verdadero"
 140 INPUT a$(k,h)
 142 PRINT a$(k,h)
 145 PRINT ,,"hay algun otro err
or en este verbo?(S/N)"; INK 2
 150 IF INKEY$="" THEN GO TO 150
 160 IF INKEY$="s" THEN GO TO 10
0
 170 RETURN
 200 CLS
 205 PRINT ,,"ahora ya tienes lo
s verbos introducidos, tienes va
rias posibili-dades: 1) Almacena
rlos en cinta          2) Que te l
os pregunte.           3) Que te l
os muestre.";AT 10,10;"ELIGE"; I
NK 4
 206 INPUT h
 207 GO TO 1000*h
1000 CLS
1010 INPUT "Quieres que llame de
 alguna for-ma en especial al pr
ograma?(s/n)";s$
1020 IF s$="n" THEN GO TO 1100
1030 PRINT ,,"Como le llamo "
1035 INPUT u$
1040 GO TO 1110
1100 LET u$="verbos"
1110 PRINT ,,"le voy a llamar ";
u$
1115 SAVE u$
1120 CLS
1130 PRINT "Atencion, nunca usar
 RUN"; INK 5
1132 GO TO 0
2000 CLS
2010 PRINT "Te los voy a ir preg
untando sal-teados. Hay varias p
osibilidades",,"1) Te pregunto e
l infinitivo y    tu dices el r
esto.";TAB 0,,;"2) Te pregunto c
ualquier tiempo     y tu dices el
 resto.";AT 10,5;"Elige si te at
reves"; INK 3
2017 INPUT ,,"Si te cansas pulsa
 3 ";i
2021 PRINT ,,"Quieres que te los
 pregunte to- dos o solo unos cu
antos (1/2)"
2022 INPUT t
2024 GO SUB 4400
2025 GO TO 2000+i*100
2100 LET d=INT (RND*a)+f
2105 CLS
2110 PRINT ,,"Cuales son los tie
mpos del verbo";a$(d,1); INK 3
2115 INPUT "pasado";p$: PRINT p$
: GO SUB 5000
2120 INPUT "participio";o$: PRIN
T o$: GO SUB 5050
2130 IF p$=a$(d,2) AND o$=a$(d,3
) THEN GO TO 2170
2140 LET q$="Hay un error. vuelv
e a pensarte-lo y me lo cuentas"
: PRINT q$; INK 0: FOR b=1 TO 30
0: NEXT b: GO TO 2110
2170 PRINT ,,"O.K., como dominas
, si no cam-  bias de opinion te
 voy a seguir preguntando" : GO
SUB 2500
2180 GO TO 2100
2200 LET d=INT (RND*a)+3: LET g=
INT (RND*3)+1: CLS
2210 PRINT "Cuales son los resta
ntes tiemposdel verbo cuyo ";("i
nfinitivo" AND g=1);("pasado" AN
D g=2);("participio" AND g=3);"
es ";TAB 0;a$(d,g)
2220 IF g=1 THEN GO TO 2250: IF
g=2 THEN GO TO 2290
2230 PRINT "Introduce el infinit
ivo": INPUT k$: GO SUB 5100: PRI
NT k$
2240 PRINT "Introduce el pasado"
```

```
: INPUT p$: GO SUB 5000: PRINT p
$
2242 IF k$=a$(d,1) AND p$=a$(d,2
) THEN GO TO 2350: PRINT q$: GO
SUB 4500
2245 PRINT q$: GO TO 2230
2250 PRINT "introduce el pasado"
: INPUT p$: GO SUB 5000: PRINT p
$
2255 PRINT "Introduce el partici
pio": INPUT o$: GO SUB 5050: PRI
NT o$
2260 IF p$=a$(d,2) AND o$=a$(d,3
) THEN GO SUB 2350
2265 PRINT ,,q$: GO SUB 4500: GO
 TO 2250
2290 PRINT "Introduce el infinit
ivo": INPUT k$: PRINT k$: GO SUB
 5100
2295 PRINT "Introduce el partici
pio": INPUT o$: GO SUB 5050: PRI
NT o$
2297 IF k$=a$(d,1) AND o$=a$(d,3
) THEN GO SUB 2350
2300 PRINT ,,q$: GO SUB 4500: GO
 TO 2290
2350 PRINT "O.K., has sabido con
testarme, sino cambias de opinio
n te voy a  seguir preguntando":
 GO SUB 2500: GO TO 2200
2500 FOR l=1 TO 200
2515 IF INKEY$="3" THEN GO TO 20
0
2520 NEXT l: CLS : RETURN
3000 CLS : LET o=0
3010 PRINT "Te los voy a ir most
rando por paginas. Cuando estes
dispuesto   pulsa una tecla"
3020 IF INKEY$="" THEN GO TO 302
0
3025 CLS : LET y$="INFINITIVO  P
ASADO  PARTICIPIO  ": PRINT y$
3030 FOR f=1 TO n
3060 FOR d=1 TO 3
3070 PRINT AT (f-o*18+1),((d-1)*
10);a$(f,d): NEXT d
3080 IF f/18=INT (f/18) THEN GO
SUB 3500
3090 NEXT f
3095 GO SUB 3500
3100 PRINT "ya he terminado"; IN
K 6: GO TO 200
3500 PRINT "  PARA SEGUIR PULSA
UNA TECLA    "
3510 IF INKEY$="" THEN GO TO 351
0
3515 LET o=o+1: CLS : PRINT y$:
RETURN
4400 IF t<>1 THEN GO TO 4450
4410 LET a=n: LET f=1: RETURN
4450 PRINT ,,"Introduce el num.
del verbo ini-cial y el final. s
i quieres que te pregunte una pa
gina dime que num. de pagina seg
uida de un punto decimal y un un
o: ejm: 2.1    (pag 2)";TAB 0;"si
 no quieres paginas pulsa cualqu
ier numero"
4455 INPUT qu: IF qu-INT qu<>0 T
HEN GO TO 4480
4465 CLS : INPUT "Verbo inicial?
";a: INPUT "Verbo final?";f: RET
URN
4480 LET f=1+18*INT (qu-1): LET
a=18: RETURN
4500 FOR b=1 TO 400: NEXT b: CLS
 : RETURN
5000 LET q=LEN p$
5010 DIM z$(10-q): LET p$=p$+z$:
 RETURN
5050 LET q=LEN o$: DIM z$(10-q):
 LET o$=o$+z$: RETURN
5100 LET q=LEN k$: DIM z$(10-q):
 LET k$=k$+z$: RETURN
```

```
INFINITIVO  PASADO    PARTICIPIO
be          was       been
cut         cut       cut
have        had       had
do          did       done
dwel        dwelt     dwelt
get         got       gotten
sit         sat       sat
run         ran       run
sing        sang      sung
rose        rise      risen
forget      forgot    forgotten
put         put       put
write       wrote     written
drink       drank     drunk
hit         hit       hit
understandunderstoodunderstood
swim        swam      swum
sleep       slept     slept
   PARA SEGUIR PULSA UNA TECLA
```

```
Cuales son los tiempos del verbo
take
took
taken

O.K., como dominas, si no cam-
bias de opinion te voy a seguir
preguntando
```

```
Cuales son los tiempos del verbo
forget
forgot
forgot
Hay un error. vuelve a pensarte-
lo y me lo cuentas
```

El coloso en llamas

Ambientado en la célebre película de la que toma el nombre, este programa hará renacer en usted sus sentimientos más nobles para salvar a sus congéneres de una muerte tan segura como brutal.

Un enorme fuego se ha declarado en uno de los rascacielos más importantes de la ciudad, avanzando rápidamente hasta alcanzar los últimos pisos. Ante la impenetrable muralla de fuego, sus inquilinos no ven otra solución que lanzarse al vacío. Y es aquí donde entra usted.

Su misión consiste en recogerlos antes de que ocurra el desenlace fatal y llevarles a un peculiar hospital en el que siempre deberá entrar despacio (nunca corriendo). Apresúrese a llevar todos los que pueda al hospital, pero no se descuide... ¡se le pueden caer encima!

Notas gráficas

ABCDEFGHIJKLMNOPQRSTU

[illegible]

Autor: M.ª Belén Cortazar. 16K

```
   1 REM   SALVAMENTO
  10 FOR n=0 TO 7: READ x: POKE
USR "+"+n,x: NEXT n
  12 FOR n=0 TO 7: READ x: POKE
USR "*"+n,x: NEXT n
  14 FOR n=0 TO 7: READ x: POKE
```

8

Simpson

Este programa utiliza la regla de Simpson para hacer integrales definidas. Si lo ejecuta directamente sólo hará representaciones gráficas. Para obtener la integral teclee *RUN* 1000, introduciendo posteriormente la función a integrar (el argumento es la variable X). A continuación se pregunta el intervalo de integración que quedará después resaltada en el gráfico. Una vez realizada la gráfica se pide el número de iteraciones para resolver la integral (diez es un número normal de iteraciones).

Pruebe con los siguientes ejemplos:

Función: SIN X

Intervalo: Xo=0 X1=PI

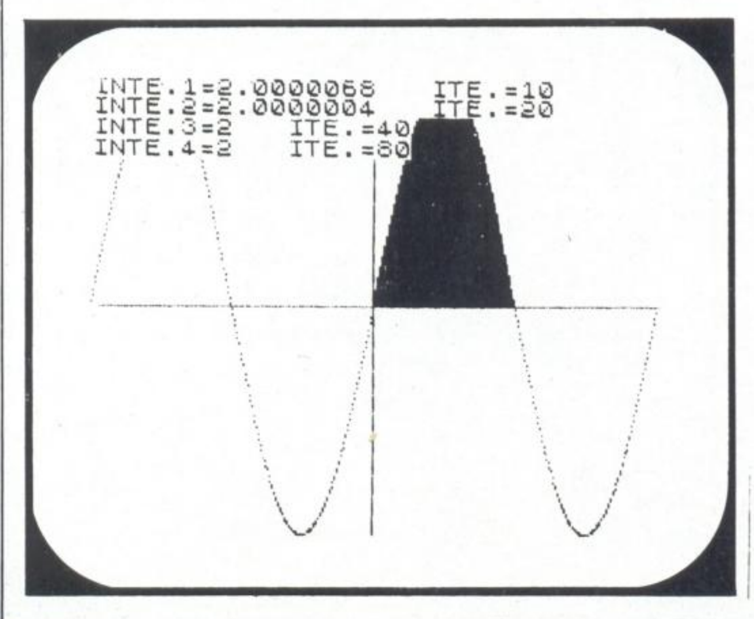
Xo=2 * PI Yo=1

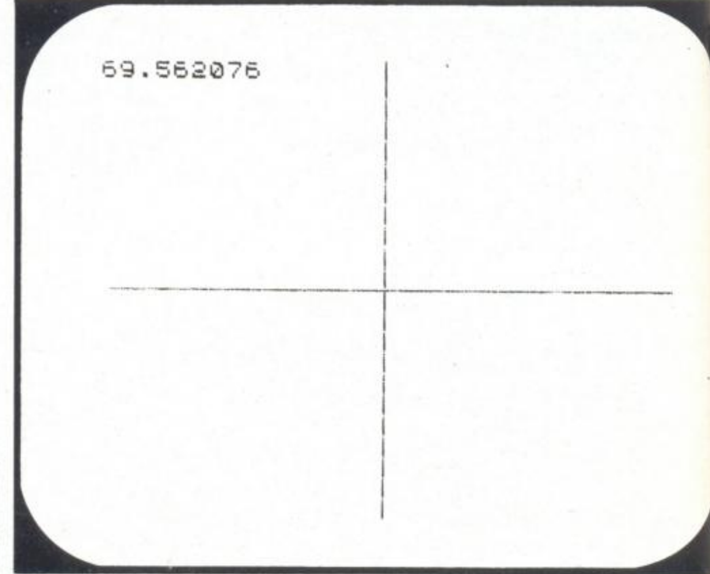
Xo=0 X1=1

Xo=2 Yo=e^2 (EXP 2)

Autor: Julio Gómez Herrero. 16K

```
   1 LET MU=0: LET r=0: LET X0=1
E30: LET X1=1E30
   5 INK 0
  10 PRINT AT 10,4;"TELCEE MAXIM
O EN ESCALA X": INPUT "Xo=";N
  15 PRINT AT 10,4;"TELCEE MAXIM
O EN ESCALA Y": INPUT "Yo=";M
  20 PRINT AT 10,1;"LA FUNCION S
E REPRESENTA EN LOS INTERVALOS:
{-Xo,Xo}/{-Yo,Yo}": PAUSE 100: C
LS
  25 PLOT 127,0: DRAW 0,175: PLO
T 0,87: DRAW 255,0
  27 IF R<>1 THEN INPUT "FUNCION
 A REPRESENTAR";A$
  30 FOR I=1E-15 TO N STEP N/255
```



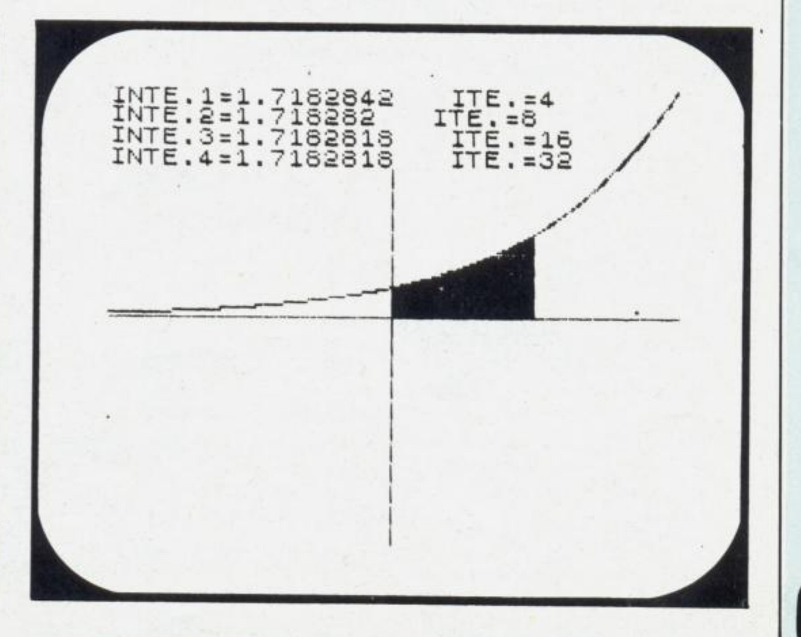


7

```
USR "■"+n,x: NEXT n
  16 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  18 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  20 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  22 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  24 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  26 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  28 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  30 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  32 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  34 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  36 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  38 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  40 FOR n=0 TO 7: READ x: POKE
USR "■"+n,x: NEXT n
  50 DATA 0,24,24,126,126,24,24,
0
  52 DATA 0,43,42,58,42,43,0,63
  54 DATA 0,187,162,187,138,186,
0,255
  56 DATA 0,174,164,164,36,36,0,
255
  58 DATA 0,232,168,232,168,174,
0,254
  60 DATA 24,24,48,60,51,40,196,
134
  62 DATA 24,24,48,60,51,48,72,1
08
  64 DATA 0,0,0,0,255,36,195,195
  66 DATA 0,192,249,63,255,36,19
5,195
  68 DATA 24,24,12,60,204,12,18,
54
  70 DATA 24,24,12,60,204,20,35,
```

```
  35 LET X=I
  40 IF  ABS (VAL A$)>M THEN PRI
NT AT 0,0;VAL A$: GO TO 60
  50 PLOT (I*127/N)+127,(VAL A$)
*87/M+87
  55 IF R=1 AND I>=X0 AND I<=X1
THEN DRAW 0,-((VAL A$)*87/M)
  60 NEXT I
  70 FOR I=1E-15 TO -N STEP -N/2
55
  75 LET X=I
  80 IF ABS (VAL A$)>M THEN PRIN
T AT 0,0;VAL A$: GO TO 100
  90 PLOT (I*127/N)+127,(VAL A$)
*87/M+87
  95 IF R=1 AND I<=X1 AND I>=X0
THEN DRAW 0,-((VAL A$)*87/M)
 100 NEXT I
 105 IF MU=1 THEN RETURN
 110 INPUT "sobreimpresion  si=1
,no=0//";res
 120 IF res=1 THEN GO TO 27
 130 IF RES=0 THEN STOP
1000 DIM X(20): INPUT "FUNCION:
";A$: LET R=1
1002 INPUT "INTERVALO ";"X0=";X0
,"X1=";X1
1004 INPUT "DESEA REPRESENTACION
 GRAFICA?:SI=1,NO=0 ";MU
1006 IF MU=1 THEN GO SUB 10
1010 BEEP 1,7: INPUT "ITERACIONE
S: P=";F
1015 LET T=F
1020 LET B=(X1-X0)/2/F
1030 LET A=0: LET X=X0: LET Y=VA
L A$
1040 LET A=Y+A: LET X=X+B: LET Y
=VAL A$
1050 LET A=Y*4+A: LET X=X+B: LET
 Y=VAL A$
1060 LET A=Y+A: LET F=F-1
1070 IF F<>0 THEN GO TO 1040
1080 LET C=A*B/3
1090 PRINT "INTE.";R;"=";C;"    I
TE.=";T: BEEP 2,7
1100 LET X(R+1)=C
1115 IF ABS (X(R+1)-X(R))>=1E-7
THEN LET R=R+1: LET T=T*2: LET F
=T: GO TO 1020
1120 FOR I=1 TO 5: BEEP .5,49: P
AUSE 5: NEXT I
```

```
74.499023
INTE.1=866.66667     ITE.=5
INTE.2=866.66667     ITE.=10
```



```
97
  72 DATA 6,38,28,78,125,4,24,16
  74 DATA 8,104,113,63,88,136,14
,2
  76 DATA 195,34,20,24,240,24,52
,50
  78 DATA 16,149,84,89,114,118,6
2,60
  80 DATA 32,164,148,197,109,111
,255,255
 100 BORDER 1: PAPER 1: INK 7: C
LS
 102 PRINT AT 0,11;"SALVAMENTO"
 104 PRINT '" Un edificio proxim
o al hospitalse incendia.Salen 2
 camilleros aintentar salvar a l
os que se ti-ran por las ventana
s."
 106 PRINT '" El juego acaba cua
ndo el fuego  alcanza el ultimo
piso,o cuando mueren los cuatro
camilleros de que dispones."
 108 PRINT '" Con la camilla car
gada,o con unsolo camillero no p
uedes recogera los que caen.Tien
es que ir al hospital para entre
gar los salvados,o para rácoger
otro camille-ro."
 109 PRINT FLASH 1;AT 21,5;"Para
 continuar pulsa P"
 110 FOR n=0 TO 27
 111 IF INKEY$="p" THEN GO TO 12
5
 112 IF n/2=INT (n/2) THEN PRINT
 AT 19,n;"     "
 113 IF n/2<>INT (n/2) THEN PRIN
T AT 19,n;"     "
 114 BEEP .2,15+n/2
 115 NEXT n
 116 FOR n=27 TO 0 STEP -1
 117 IF INKEY$="p" THEN GO TO 12
5
 118 IF n/2=INT (n/2) THEN PRINT
 AT 19,n;"     "
 119 IF n/2<>INT (n/2) THEN PRIN
T AT 19,n;"     "
 120 BEEP .2,15+n/2
 121 NEXT n
 123 GO TO 110
 125 CLS : PRINT AT 0,11;"[illegible]"
 128 PRINT ''" Teclas de movimie
nto:"''"       Q IZQUIERDA     P DER
ECHA"
 130 PRINT '"Q+CAPS SHIFT : IZQU
IERDA RAPIDA"
 132 PRINT '"P+CAPS SHIFT :  DER
ECHA  RAPIDA"
 135 PRINT '" Para desocupar la
camilla,o re-"''"coger camillero
s,PULSA [illegible] con la "''"camilla den
tro del hospital"
 138 PRINT FLASH 1;AT 18,2;" * X
 *  BUENA SUERTE   * X * "
 140 PRINT AT 21,7;"[illegible]"
 145 PAUSE 0
 150 BORDER 5: PAPER 5: INK 1: C
LS
 152 FOR n=0 TO 18: PRINT INK 4;
AT n,0;"        ": NEXT n
 154 PRINT INK 4;AT 19,0;"
 ";AT 20,1;"     "
 156 PRINT INK 7;AT 6,24;"
"
 158 FOR n=7 TO 17: PRINT INK 7;
AT n,24;"        ": NEXT n
 160 PRINT INK 7;AT 18,24;" "; I
NK 2; PAPER 7;"+HOSPITAL+"; INK 7;AT
 18,31;" "
 162 PRINT INK 7;AT 19,24;"
   "
 164 PRINT INK 7;AT 20,29;"   "
 166 PRINT AT 21,0;"

 175 LET x=4: PRINT PAPER 2; INK
 7;AT 0,19;"CAMILLEROS: 4"
 176 PRINT PAPER 2;AT 1,19;"
"
 177 LET y=0: PRINT PAPER 2; INK
 7;AT 2,19;" SALVADOS: 0 "
 178 PRINT PAPER 2;AT 3,19;"
"
 179 LET z=0: PRINT PAPER 2; INK
 7;AT 4,19;"  MUERTOS: 0 "
 200 LET a$="        "
 202 LET b$="        "
 204 LET c$="        "
 206 LET d$="        "
 208 LET e$="        "
 210 LET f$="        "
 212 LET g$="        "
 214 LET h$="        "
 216 LET i$=" "
 218 LET j$=" "
 220 LET a=25
 225 LET k=0: LET t=0
 249 IF x=0 THEN GO TO 600
 250 GO TO 251+INT (RND*3)
 251 LET k$="*": GO TO 260
 252 LET k$="X": GO TO 260
 253 LET k$="*": GO TO 260
 260 GO TO 261+INT (RND*7)
 261 LET c=.8: LET d=.7: GO TO 2
80
 262 LET c=.7: LET d=.6: GO TO 2
80
 263 LET c=.7: LET d=.5: GO TO 2
80
 264 LET c=.7: LET d=.4: GO TO 2
80
 265 LET c=.65: LET d=.25: GO TO
 280
 266 LET c=.7: LET d=.15: GO TO
280
 267 LET c=.8: LET d=0
 280 LET m=RND*7: LET n=8
 290 PRINT AT INT m,INT n;k$
 292 LET t=t+1: IF t=600 THEN GO
 TO 600
 294 IF t/2=INT (t/2) THEN LET t
$=i$: BEEP .001,40
```

SALVAMENTO

Un edificio proximo al hospital se incendia.Salen 2 camilleros a intentar salvar a los que se ti-ran por las ventanas.

El juego acaba cuando el fuego alcanza el ultimo piso,o cuando mueren los cuatro camilleros de que dispones.

Con la camilla cargada,o con un solo camillero no puedes recoger a los que caen.Tienes que ir al hospital para entregar los salvados,o para recoger otro camille-ro.

Para continuar pulsa P

```
 296 IF t/2<>INT (t/2) THEN LET
t$=j$: BEEP .001,35
 298 PRINT INK 2;AT 18,6;t$;AT 1
7,1;t$;AT 16,4;t$;AT 14,2;t$;AT
13,6;t$;AT 11,3;t$;AT 9,5;t$;AT
8,1;t$
 300 IF t>150 THEN PRINT INK 2;A
T 6,6;t$;AT 5,2;t$
 304 IF t>300 THEN PRINT INK 2;A
T 3,5;t$;AT 2,1;t$
 308 IF t>500 THEN PRINT INK 2;A
T 0,7;t$;AT 1,4;t$
 310 IF INKEY$="o" AND a>8 THEN
LET a=a-1
 311 IF INKEY$="O" AND a>9 THEN
LET a=a-2
 312 IF INKEY$="p" AND a<25 THEN
 LET a=a+1
 313 IF INKEY$="P" AND a<24 THEN
 LET a=a+2
 315 IF a/2=INT (a/2) THEN PRINT
 AT 20,a-3;a$
 320 IF a/2<>INT (a/2) THEN PRIN
T AT 20,a-3;b$
 322 IF INT m=15 THEN LET d=0
 360 IF INT m=20 AND INT n=a AND
 k=0 THEN LET k=1: LET a$=c$: LE
T b$=d$: PRINT AT 20,a-3;a$: GO
TO 250
 365 IF INT m=20 AND INT n=a+1 T
HEN LET k=2: LET a$=e$: LET b$=f
$: PRINT AT 20,a-3;a$: PRINT INV
ERSE 1;AT 21,a+1;k$: LET z=z+2:
PRINT PAPER 2; INK 7;AT 4,30;z:
GO TO 249
 368 IF INT m=20 AND INT n=a-1 T
HEN LET k=2: LET a$=g$: LET b$=h
$: PRINT AT 20,a-3;a$: PRINT INV
ERSE 1;AT 21,a-1;k$: LET z=z+2:
PRINT PAPER 2; INK 7;AT 4,30;z:
GO TO 249
 370 IF INKEY$="1" AND k=1 AND a
=25 THEN LET a$="       ": LET b
$="       ": LET k=0: LET y=y+1:
 PRINT PAPER 2; INK 7;AT 2,30;y
 375 IF INKEY$="1" AND k=2 AND a
=25 THEN LET a$="       ": LET b
$="       ": LET k=0: LET x=x-1:
 PRINT PAPER 2; INK 7;AT 0,31;x
 390 IF INT m=20 THEN PRINT INVE
RSE 1;AT 21,INT n;k$: PRINT AT 2
0,INT n;" ": LET z=z+1: PRINT PA
PER 2; INK 7;AT 4,30;z: GO TO 25
0
 400 PRINT AT INT m,INT n;" "
 410 LET m=m+c: LET n=n+d
 500 GO TO 290
 600 PRINT AT INT m,INT n;" ": F
OR n=50 TO 45 STEP -.5: BEEP .1,
n: NEXT n
 605 IF x=4 AND z<=5 THEN PRINT
FLASH 1;AT 7,11;"ENHORABUENA": P
RINT AT 9,9;"Has salvado ";y;AT
11,8;"vidas sin perder";AT 13,11
;"camilleros"
 610 IF x<4 AND z<=5 THEN PRINT
AT 7,10;"Has realizado";AT 9,9;"
una buena ope-";AT 11,9;"racion,
pero has";AT 13,9;"perdido vidas
";AT 15,8;"innecesariamente"
 615 IF z>5 THEN PRINT AT 7,10;"
Solo has sal-";AT 9,10;"vado ";y
;" vidas";AT 11,9;"?Te das cuent
a";AT 13,9;"de tu responsa-";AT
15,9;"bilidad?"
 616 IF y=1 THEN PRINT AT 9,21;"
"
 618 IF y=0 THEN PRINT AT 7,9;"N
o has salvado";AT 9,10;"ninguna
vida"
 620 PRINT AT 17,10;"PARA JUGAR
 "
 625 PRINT AT 19,8;"INSTRUCCIONE
S "
 630 FOR n=0 TO 100
 632 IF n/2=INT (n/2) THEN LET t
$=i$: BEEP .001,40
 634 IF n/4=INT (n/4) THEN LET t
$=j$: BEEP .001,35
 635 IF INKEY$="s" THEN GO TO 15
0
 636 IF INKEY$="i" THEN GO TO 10
0
 640 PRINT INK 2;AT 18,6;t$;AT 1
7,1;t$;AT 16,4;t$;AT 14,2;t$;AT
13,6;t$;AT 11,3;t$;AT 9,5;t$;AT
8,1;t$;AT 6,6;t$;AT 5,2;t$;AT 3,
5;t$;AT 2,1;t$;AT 1,4;t$;AT 0,7;
t$
 645 NEXT n
 650 GO TO 630
```

SALVAMENTO

Teclas de movimiento:

O IZQUIERDA   P DERECHA

O+CAPS SHIFT : IZQUIERDA RAPIDA

P+CAPS SHIFT : DERECHA RAPIDA

Para desocupar la camilla,o recoger camilleros,PULSA 1 con la camilla dentro del hospital

* * * BUENA SUERTE * * *

Pulsa una tecla



Circuitos eléctricos

⑨

DDD es el nombre de este programa, siglas que responden a **Decodificador De Direcciones** y gracias al cual se representa en pantalla un circuito lógico formado con cuatro puertas *NOT* y cuatro puertas *AND* que posee dos entradas, es decir, que simula un bus de dos vías y cuatro salidas.

Desde el programa se coloca en el bus la entrada 00, 01, 10 u 11, en función de lo cual se activará la salida 0, 1, 2, 3. Por los hilos se simula el movimiento de los *bits* y en las puertas *AND* aparecen los valores recibidos.

Es realmente gracioso contemplar el movimiento de los *bits* por el circuito. Muy indicado para fines didácticos y especialmente para los amantes de la circuitería.

Notas gráficas

ABCDEFGHIJKLMNOPQRSTU

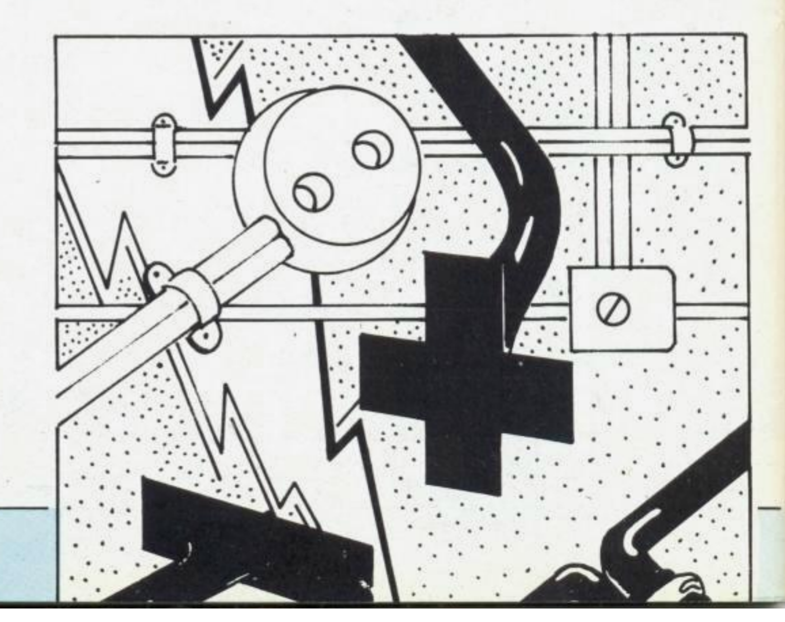
ø↓┤┬↑↓←|—JKLMNOPQRSTU

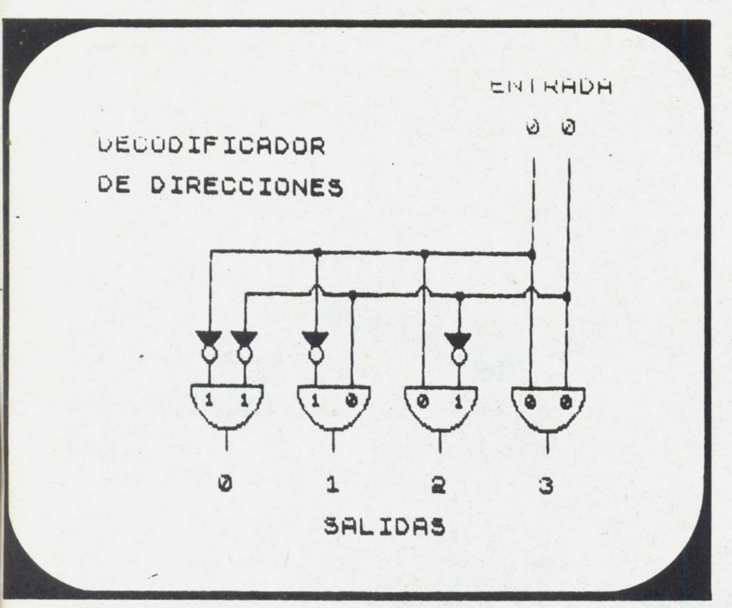
Autor: Manuel García Chumilla 16K

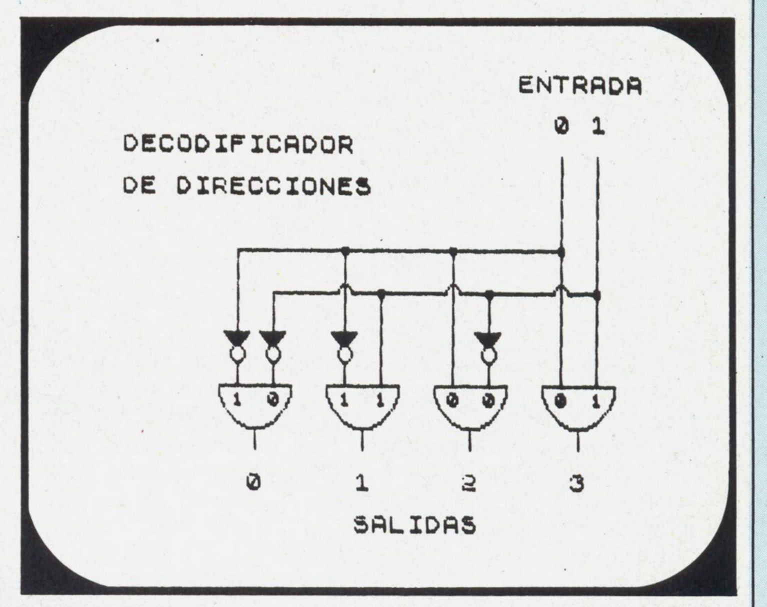
```
  10 REM      ****
  20 REM   GRAFICOS
  30 REM      ****
  40 FOR i=1 TO 9: READ a$: FOR
j=0 TO 7
  50 READ x: POKE USR a$+j,x
  60 NEXT j: NEXT i
  70 DATA "a",0,0,28,38,42,50,28
,0
  71 DATA "b",0,0,8,24,8,8,28,0
  72 DATA "c",8,8,8,28,252,28,8,
8
  73 DATA "d",0,0,0,28,255,28,8,
8
  74 DATA "e",8,28,42,73,201,8,8
,8
  75 DATA "f",8,8,8,28,28,28,8,8
  76 DATA "g",0,0,0,28,255,28,0,
0
  77 DATA "h",8,8,8,8,8,8,8,8
  78 DATA "i",0,0,0,0,255,0,0,0
  80 REM            ****
  90 REM Trazado del circuito
 100 REM            ****
 110 PLOT 212,143: DRAW 0,-87: P
LOT 212,107: DRAW -48,0: DRAW 0,
-51: DRAW 0,51: DRAW -48,0: DRAW
 0,-30: GO SUB 1000: DRAW 0,30:
DRAW -48,0: DRAW 0,-30: GO SUB 1
000
 120 PLOT 228,143: DRAW 0,-87: P
LOT 228,91: DRAW -48,0: PRINT AT
 10,26;"↑": DRAW 0,-14: GO SUB 1
000: DRAW 0,14: DRAW -48,0: PRIN
T AT 10,20;"↑": DRAW 0,-35: DRAW
 0,35: DRAW -48,0: PRINT AT 10,1
4;"↑": DRAW 0,-14: GO SUB 1000
 130 FOR i=1 TO 4: PLOT 48*i+12,
56: DRAW 32,0: DRAW -16,-16,-PI/
2: DRAW 0,-8: DRAW 0,8: DRAW -16
,16,-PI/2: NEXT i
 140 PRINT AT 0,24;"ENTRADA";AT
3,2;"DECODIFICADOR";AT 5,2;"DE D
IRECCIONES";AT 21,15;"SALIDAS"
 150 PRINT #1;AT 1,0;"Entrada iz
quierda:  Pulse 0/1 "
 160 LET b=CODE INKEY$: IF b<48
OR b>49 THEN GO TO 160
 170 PRINT AT 2,26;CHR$ b: FOR i
=0 TO 50: NEXT i: PRINT #1;AT 1,
0;"

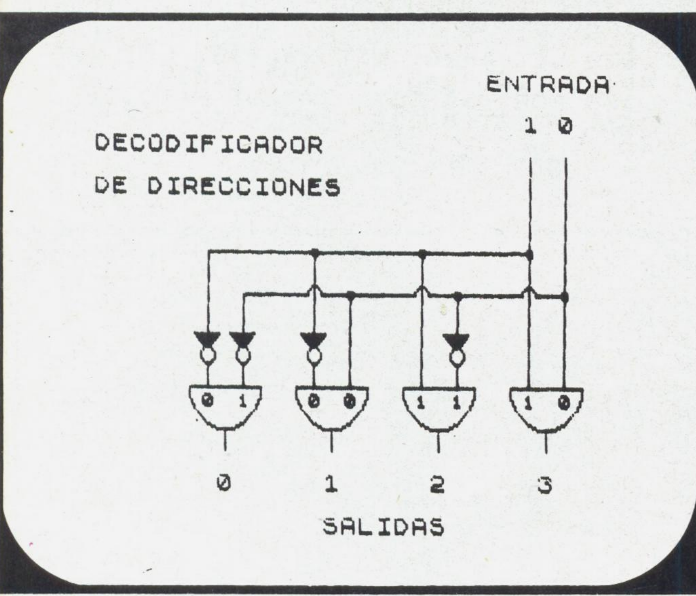
 180 PRINT #1;AT 1,0;"Entrada de
recha:  Pulse 0/1
 190 LET a=CODE INKEY$: IF a<48
OR a>49 THEN GO TO 190
 200 PRINT AT 2,28;CHR$ a: PRINT
 #1;AT 1,0;"

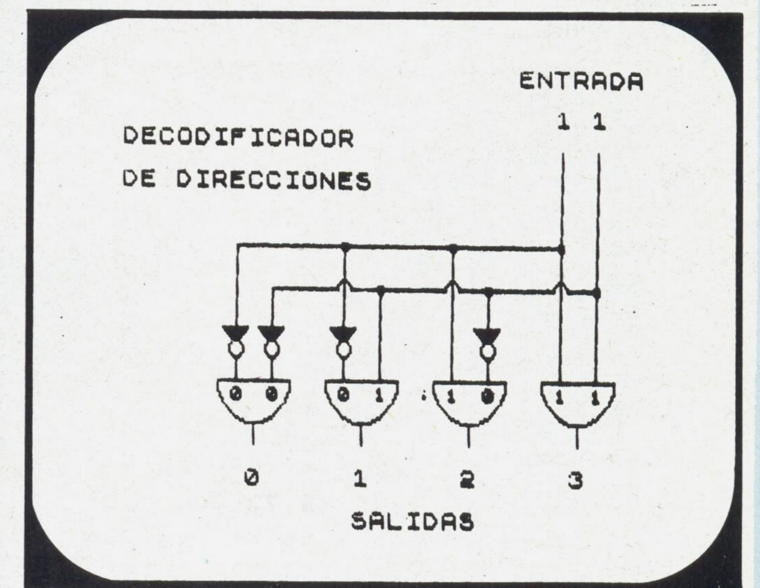
 210 REM      ***
 220 REM Movimiento
 230 REM      ***
 240 FOR i=1 TO 10: PRINT AT 3+i
,28;"↓";AT 3+i,26;"↓": PAUSE 20:
 PRINT AT 3+i,28;"|";AT 3+i,26;"
|"
 242 IF i=5 THEN PRINT AT 8,26;"
┤";AT 8,28;"|"
 250 IF i=7 THEN PRINT AT 10,26;
"↑";AT 10,28;"┤"
 260 NEXT i
 270 LET p=28: LET x=a+96: LET y
=b+96: LET q=3: GO SUB 1100
 300 FOR i=1 TO 5: PRINT AT 10,2
8-i;"←";AT 8,26-i;"←": PAUSE 20:
 PRINT AT 10,28-i;"—";AT 8,26-i;
"—"
 310 IF i=2 THEN PRINT AT 10,26;
"↑"
 320 NEXT i
 330 PRINT AT 8,20;"┬";AT 10,22;
"┬"
 340 FOR i=1 TO 5: PRINT AT 8+i,
20;"↓": PAUSE 20: PRINT AT 8+i,2
0;"|"
 350 IF i=2 THEN PRINT AT 10,20;
"↑"
 352 IF i=3 THEN PRINT AT 11,22;
"↓"
 354 IF i=4 THEN PRINT AT 11,22;
"|"
 360 NEXT i
 370 LET x=144: IF a=48 THEN LET
 x=145
 380 LET p=22: LET y=b+96: LET q
=2: GO SUB 1100
 390 FOR i=1 TO 5: PRINT AT 10,2
2-i;"←";AT 8,20-i;"←": PAUSE 20:
 PRINT AT 10,22-i;"—";AT 8,20-i;
"—"
 400 IF i=2 THEN PRINT AT 10,20;
"↑"
 410 NEXT i
 420 PRINT AT 8,14;"┬";AT 10,16;
```












```
"↓"
 430 FOR i=1 TO 3: PRINT AT 8+i,
14;"↓";AT 10+i,16;"↓": PAUSE 20:
 PRINT AT 8+i,14;"|";AT 10+i,16;
"|"
 440 IF i=2 THEN PRINT AT 10,14;
"↑"
 450 NEXT i
 460 LET y=144: IF b=48 THEN LET
 y=145
 470 LET p=16: LET q=1: LET x=a+
96: GO SUB 1100
 480 FOR i=1 TO 5: PRINT AT 10,1
6-i;"←";AT 8,14-i;"←": PAUSE 20:
 PRINT AT 10,16-i;"—";AT 8,14-i;
"—"
 482 IF i=2 THEN PRINT AT 10,14;
"↑"
 484 NEXT i
 490 FOR i=1 TO 2: PRINT AT 8+i,
8;"↓": PAUSE 20: PRINT AT 8+i,8;
"|": NEXT i
 500 PRINT AT 11,8;"↓";AT 11,10;
"↓": PAUSE 20: PRINT AT 11,8;"|"
;AT 11,10;"|"
 510 LET x=144: IF a=48 THEN LET
 x=145
 520 LET y=144: IF b=48 THEN LET
 y=145
 530 LET p=10: LET q=0: GO SUB 1
100
 540 PRINT #1;AT 1,0;"Hacemos ot
ra prueba ?     (s/n)"
 550 IF INKEY$="" THEN GO TO 550
 560 IF INKEY$="s" THEN GO SUB 1
200: GO TO 150
 570 CLS : STOP
1000 DRAW 0,-1: DRAW -6,0: LET p
=1
1010 FOR j=1 TO 5: DRAW p*(13-2*
j),0: LET p=-p: DRAW p,-1: NEXT
j
1020 DRAW 0,-6,PI: DRAW 0,-9: DR
AW 0,9: DRAW -0,6,PI: DRAW 0,6:
RETURN
1100 PRINT AT 15,p;CHR$ x;AT 15,
p-2;CHR$ y
1110 IF x+y=290 THEN FLASH 1
1120 PRINT AT 19,p-1;q: FLASH 0:
 RETURN
1200 FOR i=1 TO 4: PRINT AT 19,6
*i+3;" ": PRINT AT 15,6*i+2;"
": NEXT i: PRINT AT 2,26;"    ":
RETURN
```

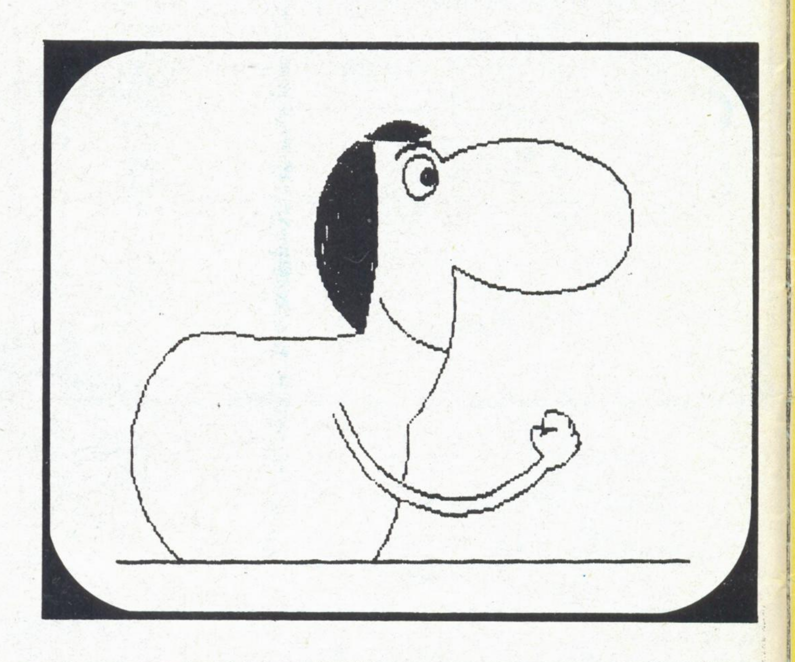
gusánez *por José C. Tomás*



Con el nacimiento de Todospectrum asistimos también al del intrépido Gusanez, una mascota peculiar que pondrá la nota de humor en las difíciles, complicadas y a veces pesadas operaciones con el Spectrum. Deje que nuestro Gusanez, el de todos, irrumpa en su monitor o televisor con este pequeño y sencillo programa.

A Gusanez le van a ocurrir todo tipo de aventuras, unas inverosímiles, otras arriesgadas, pero será el humor la nota predominante de las acciones que protagonizará gusanez en esta última página de Todospectrum. Nuestra mascota está hecha para divertir, su creador es un joven enamorado del Spectrum, al que ha hecho trabajar duramente para dar cuerpo a su obra.

```
  10 PLOT 0,0: DRAW 255,0
  15 PLOT 30,0: DRAW 0,85,-2
  20 DRAW 20,2,-.4
  25 DRAW 60,0,.3
  30 DRAW 0,70,-2
  35 DRAW 30,0,-1
  40 DRAW 5,-5,-1
  45 DRAW 75,-3,-1
  50 DRAW 10,-20,-1
  55 DRAW -10,-20,-1
  60 DRAW -70,3,-1
  65 DRAW -20,-60,-1
  70 DRAW -2,-21,-.5
  71 PLOT 126,23: DRAW -11,-22,-
.6
  75 CIRCLE 135,146,8
  80 FOR n=1 TO 3: CIRCLE 139,14
6,n: NEXT n
  85 PLOT 125,154: DRAW 15,5,-1
  90 PLOT 125,153: DRAW 15,5,-1
  95 FOR n=2.2 TO -.5 STEP -.1:
PLOT 110,87: DRAW 3,75,-n: NEXT
n
 100 FOR n=1.5 TO -.3 STEP -.1:
PLOT 113,160: DRAW 27,3,-n: NEXT
 n
 105 PLOT 148,80: DRAW -30,20,-1
 110 PLOT 100,60: DRAW 90,-20,2
 115 DRAW -5,10,-1
 120 DRAW 5,0,-2
 125 PLOT 189,48: DRAW 7,2,-1
 130 PLOT 191,50: DRAW 6,6,-3
 135 DRAW 5,-5,-2
 140 PLOT 203,53: DRAW 0,-5,-2
 145 PLOT 205,49: DRAW 0,-5,-2
 150 PLOT 207,45: DRAW -15,-10,-
1
 155 DRAW -95,21,-2
```









La versión española de Popular Computing

ORDENADOR POPULAR

LA REVISTA QUE INTERESA TANTO AL AFICIONADO COMO AL PROFESIONAL





Una publicación que informa con amenidad acerca de las novedades en el campo de las computadoras personales.

ORDENADOR POPULAR, la revista para el aficionado a la informática.

Ya está a la venta

Cómprela en su kiosco habitual o solicítela a:

ORDENADOR POPULAR

Bravo Murillo, 377
Tel. 7339662
28020 - MADRID

16 K



GARANTIZAMOS EL CRECIMIENTO DE TU SPECTRUM

Si tienes un SINCLAIR ZX SPECTRUM de 16 K y deseas ampliarlo a 48 K ahora puedes hacerlo con toda **garantía.**
Acude a tu Concesionario Autorizado INVESTRONICA y en breve espacio de tiempo dispondrás de tu Spectrum con MAS POTENCIA.
Además INVESTRONICA ampliará por tres meses la garantía de tu equipo, independientemente de la fecha de adquisición y te obsequiará con una Cinta de Demostración de 48 K.

IMPORTANTE:
Al adquirir los productos **SINCLAIR** exija la TARJETA DE GARANTIA INVESTRONICA, única válida en todo el territorio nacional y llave para cualquier resolución de duda o reparación. INVESTRONICA no prestará ningún servicio técnico a todos aquellos aparatos que carezcan de la correspondiente garantía.
DE VENTA EN CONCESIONARIOS AUTORIZADOS.



48 K

Amplía, ahora, tu SPECTRUM de 16 K a 48 K.

Con garantía...
Con más garantía.



J. M. PUBLICIDAD